PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-337864
(43)Date of publication of application : 07.12.2001

(51)Int.Cl. G06F 12/14
G06F 1/00
G06F 12/00
G06F 15/00

(21)Application number : 2000-300561
(22)Date of filing : 28.09.2000

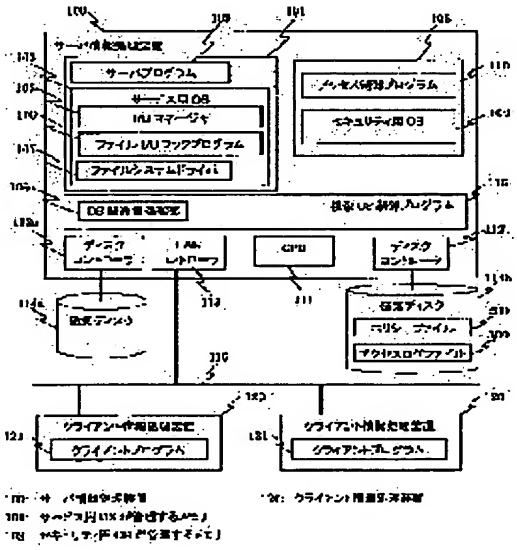
(71)Applicant : HITACHI LTD
(72)Inventor : ARAI MASATO
MATSUMOTO HITAKA
KAJIURA TOSHINORI

(30)Priority
Priority number : 2000084706 Priority date : 22.03.2000 Priority country : JP

(54) ACCESS CONTROL SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide an access control system and the method of the same by which illegal access can be inhibited, even if any intruder from a network attempts illegal readout or writing on files by abusing a user's authority.
SOLUTION: A policy is employed, that access to specific files is permitted only when a specific user uses a specific program. And further, the access control is performed based on the policy by registering the policy in a policy file 200, and passing access informtaion hooked by a file I/O hooking program 106 to an access control program 110 on a OS for security 104 through a communication process division between OSs 108.

図1



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－337864
（P2001－337864A）
(43)公開日　平成13年12月７日(2001.12.7)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
Ｇ０６Ｆ　12/14	３１０	Ｇ０６Ｆ　12/14	３１０Ｋ	５Ｂ０１７
1/00	３７０	1/00	３７０Ｅ	５Ｂ０７６
		12/00	５３７Ｍ	５Ｂ０８２
12/00	５３７	15/00	３３０Ａ	５Ｂ０８５
15/00	３３０	9/06	６６０Ｅ	

審査請求　未請求　請求項の数38　ＯＬ　（全 31 頁）

(21)出願番号　特願2000－300561(P2000－300561)
(22)出願日　平成12年９月28日(2000.9.28)
(31)優先権主張番号　特願2000－84706(P2000－84706)
(32)優先日　平成12年３月22日(2000.3.22)
(33)優先権主張国　日本（ＪＰ）

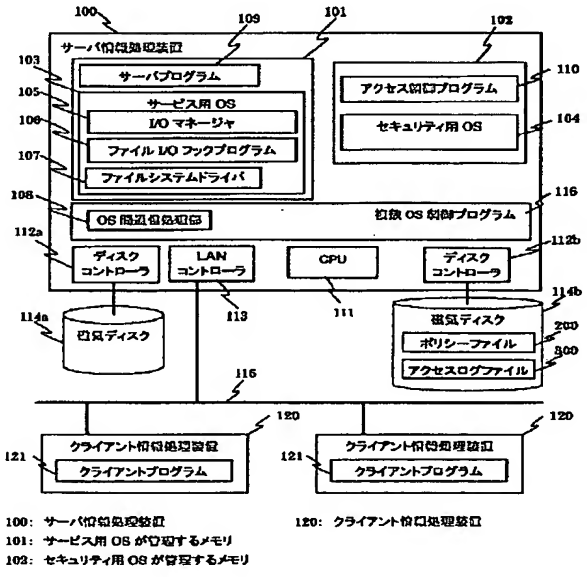
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　荒井　正人
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(72)発明者　松本　日高
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(74)代理人　100075096
弁理士　作田　康夫

最終頁に続く

(54)【発明の名称】　アクセス制御システム

(57)【要約】
【課題】ネットワークからの侵入者がいかなるユーザー権限を利用して不正なファイル読み出しや書き込みを試みても、該アクセスの抑止が可能なアクセス制御システム及びその方法を提供する。
【解決手段】特定のファイルへのアクセスを、特定のユーザーが特定のプログラムを使用した場合に限り許可するといったポリシーを用いる。さらに、ポリシーをポリシーファイル２００に登録し、ファイルＩ／Ｏフックプログラム１０６がフックしたアクセス情報を、ＯＳ間通信処理部１０８を介してセキュリティ用ＯＳ１０４上のアクセス制御プログラム１１０に渡し、前記ポリシーに基づいたアクセス制御を行う。

図１



【特許請求の範囲】

【請求項１】各種処理に必要な情報を格納するための記憶部と、
記憶媒体に格納されたファイルへの入出力処理部と、
前記ファイルへのアクセス制御ポリシーを記述した，前記記憶部中のポリシーファイルと、
当該ファイルアクセス要求の正当性を前記アクセス制御ポリシーに基づき判定するアクセス制御処理部と、
前記入出力処理部を用いたファイルアクセス要求の発行を監視し、発行されたファイルアクセス要求を前記アクセス制御処理部に伝え、正当性判定結果を前記アクセス制御処理部から受信する監視処理部と、
前記入出力処理部とアクセス制御処理部と監視処理部各々が使用する前記記憶部中の記憶領域と，前記ポリシーファイルとを、前記前記入出力処理部とアクセス制御処理部と監視処理部のいずれとも異なるアクセス実行処理部から保護するための排他制御部と、を備え、
前記ポリシーファイルは、アクセス制御ポリシーとして，アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行処理部とアクセスタイプとを特定する情報を備え、
前記監視処理部は、アクセス要求発行元とアクセス実行処理部とアクセスタイプとを特定する情報を用いて、前記発行されたファイルアクセス要求を伝えるアクセス制御システム。

【請求項２】請求項１記載のアクセス制御システムであって、
前記アクセス制御ポリシーは、前記ファイルに対して禁止されたアクセスタイプと、該禁止されたアクセスが発生した場合にアクセス要求発行元に返すべきエラーコードと、例外として該アクセス要求を許可されたアクセス実行処理部とアクセス要求発行元とを特定する情報とからなるアクセス制御システム。

【請求項３】請求項２記載のアクセス制御システムであって、
前記アクセス制御ポリシーに記すアクセス実行処理部は、プログラムであり、該プログラムのパス名と、該プログラムの特徴値との組合せで特定するアクセス制御システム。

【請求項４】請求項３記載のアクセス制御システムにおいて、
さらに、ファイルアクセス要求内容を登録するアクセスログファイルを備え、前記アクセス制御処理部は、前記ファイルアクセス要求を前記ポリシーファイルの記述と照合し、正当性の判定結果を前記監視処理部に送信するとともに、
前記アクセス要求が許可されるものであれば、当該アクセス実行処理部の特徴値を、前記監視処理部に送信し、
前記アクセス要求が前記アクセス制御ポリシーに違反するものであれば、該ファイルアクセス要求内容を、前記アクセスログファイルに登録するアクセス制御システム。

【請求項５】請求項４記載のアクセス制御システムにおいて、
さらにオープンファイルテーブルを備え、
前記アクセス制御処理部により正当であると判定された前記ファイルアクセス要求のアクセスタイプがオープンアクセスである場合、
前記アクセス制御処理部からレスポンス情報として取得した、アクセスタイプと、前記アクセス対象となるファイルと、前記アクセス要求発行元と、アクセス実行処理部とを特定する情報を、前記オープンファイルテーブルに登録する処理部と、
前記アクセス要求としてリードアクセスまたはライトアクセスが発行された場合は、
前記オープンファイルテーブルを検索し、前記アクセス要求の正当性を判定する処理部を備えるアクセス制御システム。

【請求項６】請求項５記載のアクセス制御システムであって、
前記監視処理部は、さらに、
前記オープンファイルテーブルに登録されていないリードアクセス要求またはライトアクセス要求を検知した場合は、前記アクセス制御処理部を介して当該アクセス要求内容を前記アクセスログファイルに登録する処理部を備えるアクセス制御システム。

【請求項７】請求項６記載のアクセス制御システムであって、
前記監視処理部は、さらに、
ファイルクローズ要求を検知した場合には前記オープンファイルテーブルから該当する情報を削除する処理部を備えるアクセス制御システム。

【請求項８】請求項４記載のアクセス制御システムにおいて、
前記アクセス制御処理部は、
当該属性情報と前記ポリシーファイルの記述との前記照合を、前記ファイルアクセス要求のアクセスタイプがオープンアクセスであり、且つ当該ファイルアクセス要求の属性情報として、リードアクセスまたはライトアクセスに関する情報が含まれる場合におこなうアクセス制御システム。

【請求項９】請求項８記載のアクセス制御システムであって、
前記監視処理部は、さらに、
前記ファイルアクセス要求が正当であると判定された場合、前記前記監視処理部は前記アクセス実行処理部の特徴値を算出し、前記アクセス制御処理部から受信した前記特徴値とを比較する処理部と、
一致した場合には前記アクセス要求を許可する処理部と、

一致しない場合には前記ファイルアクセス要求を無効にすると共に、前記アクセス制御処理部を介して当該ファイルアクセス内容を前記アクセスログファイルに登録する処理部とを備えるアクセス制御システム。

【請求項１０】ファイルへの入出力処理部と排他的に使用する第１の記憶処理部を管理する第１のＯＳと、排他的に使用する第２の記憶処理部を管理する第２のＯＳと、前記第１のＯＳと第２のＯＳとの間でデータ通信するための通信処理部とを具備した情報処理システムにおいて、

前記第１のＯＳは，管理する前記ファイル入出力処理部へ発行されたファイルアクセス要求を監視する監視処理部を備え，前記第２のＯＳは，前記ファイルアクセス要求の正当性をアクセス制御ポリシーに基づき判定するアクセス制御処理部を備え，前記監視処理部は、前記アクセス要求を、前記通信処理部を介して、前記アクセス制御処理部に伝え、正当性判定結果を、前記通信処理部を介して、前記アクセス制御処理部から受信する情報処理システム。

【請求項１１】請求項１０記載の情報処理システムにおいて、

前記第２のＯＳは、前記アクセス制御ポリシーとして、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行処理部とアクセスタイプとを特定する情報を記すポリシーファイルを備える情報処理システム。

【請求項１２】請求項１１記載の情報処理システムであって、

前記アクセス制御ポリシーに記すアクセス実行処理部は、プログラムであり、該プログラムのパス名と、該プログラムの特徴値との組合せで特定する情報処理システム。

【請求項１３】請求項１２記載の情報処理システムにおいて、

前記第２のＯＳは、さらに、ファイルアクセス要求内容を登録するアクセスログファイルを備え、

前記アクセス制御処理部は、

前記ファイルアクセス要求を前記ポリシーファイルの記述と照合し、正当性の判定結果を前記監視処理部に送信するとともに、

前記アクセス要求が許可されるものであれば、当該アクセス実行処理部の特徴値を、前記監視処理部に送信し、

前記アクセス要求が前記アクセス制御ポリシーに違反するものであれば、該ファイルアクセス要求内容を、前記アクセスログファイル登録する情報処理システム。

【請求項１４】請求項１３記載の情報処理システムにおいて、

前記第１のＯＳは、さらにオープンファイルテーブルを備え、

前記アクセス制御処理部により正当であると判定された前記ファイルアクセス要求のアクセスタイプがオープンアクセスである場合、

前記アクセス制御処理部からレスポンス情報として取得した、アクセスタイプと、前記アクセス対象となるファイルと、前記アクセス要求発行元と、アクセス実行処理部とを特定する情報を、前記オープンファイルテーブルに登録する処理部と、

前記アクセス要求としてリードアクセスまたはライトアクセスが発行された場合は、

前記オープンファイルテーブルを検索し、前記アクセス要求の正当性を判定する処理部を備える情報処理システム。

【請求項１５】請求項１４記載の情報処理システムであって、

前記監視処理部は、さらに、

前記オープンファイルテーブルに登録されていないリードアクセス要求またはライトアクセス要求を検知した場合は、前記アクセス制御処理部を介して当該アクセス要求内容を前記アクセスログファイルに登録する処理部を備える情報処理システム。

【請求項１６】請求項１５記載の情報処理システムであって、

前記監視処理部は、さらに、

ファイルクローズ要求を検知した場合には前記オープンファイルテーブルから該当する情報を削除する処理部を備える情報処理システム。

【請求項１７】請求項１３記載の情報処理システムであって、

前記アクセス制御処理部は、当該属性情報と前記ポリシーファイルの記述との前記照合を、前記ファイルアクセス要求のアクセスタイプがオープンアクセスであり、且つ当該ファイルアクセス要求の属性情報として、リードアクセスまたはライトアクセスに関する情報が含まれる場合に行う情報処理システム。

【請求項１８】請求項１７記載の情報処理システムであって、

前記監視処理部は、さらに、

前記ファイルアクセス要求が正当であると判定された場合、前記前記監視処理部は前記アクセス実行処理部の特徴値を算出し、前記アクセス制御処理部から受信した前記特徴値とを比較する処理部と、

一致した場合には前記アクセス要求を許可する処理部と、

一致しない場合には前記ファイルアクセス要求を無効にすると共に、前記アクセス制御処理部を介して当該ファイルアクセス内容を前記アクセスログファイルに登録する処理部とを備える情報処理システム。

【請求項１９】ファイルへの入出力処理部と排他的に使用する第１の記憶処理部を管理する第１のＯＳと、排他的に使用する第２の記憶処理部を管理する第２のＯＳ

と、前記第１のＯＳと第２のＯＳとの間でデータ通信するための通信処理部とを具備した情報処理システムに用いるアクセス制御システムであって、
前記第１のＯＳに組み込み、管理する前記ファイル入出力処理部へ発行されたファイルアクセス要求を監視する監視処理部と，前記第２のＯＳが制御し、前記ファイルアクセス要求の正当性をアクセス制御ポリシーに基づき判定するアクセス制御処理部と、を備え、
前記監視処理部は、前記アクセス要求を、前記通信処理部を介して、前記アクセス制御処理部に伝え、正当性判定結果を、前記通信処理部を介して、前記アクセス制御処理部から受信する情報処理システム。
【請求項２０】ファイルへの入出力処理部と、前記ファイルを格納する記憶処理部とを備えた情報処理装置に読み込まれ、実行され、前記情報処理装置上にアクセス制御システムを構成させるプログラムと前記プログラムが使用するファイルとを格納した記憶媒体であって、
前記プログラムが使用するファイルは、アクセス制御ポリシーを記述したポリシーファイルであって、
前記アクセス制御ポリシーは、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行処理部とアクセスタイプとを特定する情報を記したものであり、
前記プログラムは、アクセス制御プログラムと、監視プログラムとを備え、前記アクセス制御プログラムは、前記情報処理装置に、
当該ファイルアクセス要求の正当性を前記アクセス制御ポリシーに基づき判定させ、
前記監視プログラムは、前記情報処理装置に、
前記入出力処理部を用いたファイルアクセス要求の発行を監視させ、
発行されたファイルアクセス要求を、アクセス要求発行元とアクセス実行処理部とアクセスタイプとを特定する情報を用いて、前記アクセス制御プログラムに伝えさせることを特徴とする記憶媒体。
【請求項２１】各種処理に必要な情報を格納するための記憶部と、　サービス提供の役割を有する第１のプログラムと前記第１のプログラムのサービスを利用する第２のプログラムとが相互に通信するためのプロセス間通信処理部と、
前記第１のプログラムと第２のプログラムとを特定する情報と，前記第１のプログラムと第２のプログラムとによるプロセス間通信に関するアクセス制御ポリシーを記述した，前記記憶部中の通信制御ポリシーファイルと、
当該プロセス間通信要求の正当性を前記通信制御ポリシーに基づき判定するアクセス制御処理部と、
前記第１のプログラムから前記プロセス間通信処理部へ発行される第１の通信要求通信要求を、前記第１のプログラムを特定する情報を用いて前記アクセス制御処理部に伝え、前記第１の通信要求の正当性判定結果を前記アクセス制御処理部から受信する第１の監視処理部と、
前記第２のプログラムから前記プロセス間通信処理部へ発行される第２の通信要求を、前記第１のプログラムを特定する情報を用いて前記アクセス制御処理部に伝え、
前記第２の正当性判定結果を前記アクセス制御処理部から受信する第２の監視処理部と、
前記プロセス間通信処理部と前記アクセス制御処理部と前記第１の監視処理部と前記第２の監視処理部が各々利用する記憶領域と、および前記通信制御ポリシーファイルを、前記プロセス間通信処理部とアクセス制御処理部と第１の監視処理部と第２の監視処理部のいずれにも該当しないアクセス実行処理部から保護するための排他制御部と、を備えるアクセス制御システム。
【請求項２２】請求項２１記載のアクセス制御システムであって、
前記第１のプログラムを特定する情報は、前記第１のプログラムを一意に識別可能な情報と、サービスの種類を識別するためのサービス番号との組合せであり、
前記第２のプログラムを特定する情報は、前記第２のプログラムを一意に識別可能な情報と、該プログラムの利用者情報と、特徴値との組合せであるアクセス制御システム。
【請求項２３】請求項２２記載のアクセス制御システムにおいて、
さらに、プロセス間通信処理を実行中のプログラムの情報を登録する通信管理テーブルを備え、
前記アクセス制御処理部は、
現在プロセス間通信可能な状態にある前記第１のプログラムを特定する情報と、該第１のプログラムとプロセス間通信処理中の前記第２のプログラムを特定する情報とを、前記通信管理テーブルに登録して保持し、
前記第２の監視処理部からプロセス間通信要求を伝えられた場合は、前記通信管理テーブルを検索し、第１のプログラムがプロセス間通信可能な状態にあり、且つ該第１のプログラムとプロセス間通信処理中の第２のプログラムが存在するか否かを判定することを特徴とするアクセス制御システム。
【請求項２４】請求項２３記載のアクセス制御システムにおいて、
前記アクセス制御処理部は、現在プロセス間通信可能な状態にある第1のプログラムを特定する情報として、該第１のプログラムを一意に識別可能な情報と、サービスの種類を識別するためのサービス番号と、該第１のプログラムが指定した暗証情報とを前記通信管理テーブルに登録し、
該第１のプログラムとプロセス間通信処理中の第２のプログラムを特定する情報として、該第２のプログラムを一意に識別可能な情報を前記通信管理テーブルに登録して保持することを特徴とするアクセス制御システム。
【請求項２５】請求項２４記載のアクセス制御システム

において、
さらに、プロセス間通信要求内容を登録する通信ログファイルを備え、
前記アクセス制御処理部は、
前記第１の監視処理部から伝えられる，前記第１のプログラムによるプロセス間通信要求を前記通信制御ポリシーファイルの記述と照合し、該プロセス間通信要求が前記通信制御ポリシーに登録されていれば、該第１のプログラムを特定する情報を、前記通信管理テーブルに登録する処理部と、
前記第１のプログラムによる該プロセス間通信要求が前記通信制御ポリシーに登録されていなければ、該プロセス間通信要求内容を、前記通信ログファイルに登録する処理部と、
前記第２の監視処理部から伝えられる，前記第２のプログラムによるプロセス間通信要求を前記通信制御ポリシーファイルの記述と照合し、該プロセス間通信要求が前記通信制御ポリシーにて許可されており、かつ前記第１のプログラムと通信可能な状態であれば、該第２のプログラムを特定する情報を前記通信管理テーブルに登録して、前記暗証情報を通信管理テーブルから取り出して前記第２の監視処理部に返信する処理部と、
前記第２のプログラムによる該プロセス間通信要求が前記通信制御ポリシーに登録されていなければ、該プロセス間通信要求内容を、前記通信ログファイルに登録する処理部と、を備えることを特徴とするアクセス制御システム。
【請求項２６】請求項２５記載のアクセス制御システムであって、
前記アクセス制御処理部は、さらに、
前記第１の監視処理部または第２の監視処理部からプロセス間通信の終了要求を伝えられた場合には、前記通信管理テーブルから該当する情報を削除する処理部を備えるアクセス制御システム。
【請求項２７】請求項２６記載のアクセス制御システムであって、
前記第１の監視処理部は、
前記第１のプログラムからプロセス間通信要求が発行された場合に、該第１のプログラムを特定する情報を取得してから、前記発行されたプロセス間通信要求を前記アクセス制御処理部に伝えるアクセス制御システム。
【請求項２８】請求項２７記載のアクセス制御システムであって、
前記第２の監視処理部は、
前記第２のプログラムからプロセス間通信要求が発行された場合に、該第２のプログラムを特定する情報を取得してから、前記プロセス間通信の開始要求を前記アクセス制御処理部に伝えるアクセス制御システム。
【請求項２９】排他的に使用する第１の記憶処理部を管理する第１のＯＳと、排他的に使用する第２の記憶処理部を管理する第２のＯＳと、前記第１のＯＳと第２のＯＳとの間でデータ通信するための通信処理部とを具備した情報処理システムにおいて、
前記第１のＯＳは，前記通信処理部へ発行されたプロセス間通信要求を監視する第１の監視処理部を備え，前記第２のＯＳは，前記プロセス間通信要求の正当性を通信制御ポリシーに基づき判定するアクセス制御処理部を備え，前記第１の監視処理部は、前記第１のＯＳ上で動作する第１のプログラムを特定する情報を用いて、前記発行されたプロセス間通信要求を、前記通信処理部を介して前記アクセス制御処理部に伝え、正当性判定結果を、前記通信処理部を介して、前記アクセス制御処理部から受信する情報処理システム。
【請求項３０】請求項２９記載の情報処理システムにおいて、
さらに、排他的に使用する第３の記憶処理部を管理する第３のＯＳを備え、
前記第３のＯＳは、通信処理部へ発行されたプロセス間通信要求を監視する第２の監視処理部を備え、
前記第２の監視処理部は、第３のＯＳ上で動作する第２のプログラムを特定する情報を用いて、前記発行されたプロセス間通信要求を、前記通信処理部を介して、前記アクセス制御処理部に伝え、正当性判定結果を前記通信処理部を介して、前記アクセス制御処理部から受信する情報処理システム。
【請求項３１】請求項３０記載の情報処理システムであって、
前記第２のＯＳは、前記通信制御ポリシーとして、前記プロセス間通信が可能な第１のプログラムを特定する情報と、該第１のプログラムとの通信が許可された第２のプログラムを特定する情報との組合せを記すポリシーファイルを備えることを特徴とする情報処理システム。
【請求項３２】請求項３１記載の情報処理システムであって、
前記通信制御ポリシーに記す第１のプログラムを、該プログラムのパス名と、サービスの種類を識別するためのサービス番号との組合せで特定し、
前記通信制御ポリシーに記す第２のプログラムを、該プログラムのパス名と、該プログラムの利用者情報と、特徴値との組合せで特定することを特徴とする情報処理システム。
【請求項３３】請求項３２記載の情報処理システムにおいて、
前記通信処理部は、前記第１のＯＳと第２のＯＳと第３のＯＳとの間で行われるデータ通信を制御し、
前記第２のＯＳは、さらに、プロセス間通信処理を実行中のプログラムの情報を登録する通信管理テーブルを備え、
前記アクセス制御処理部は、現在プロセス間通信可能な状態にある第１のプログラムを特定する情報と、該第１

のプログラムとプロセス間通信処理中の前記第３のＯＳ上で動作する第２のプログラムを特定する情報とを、前記通信管理テーブルに登録して保持し、
前記第２の監視処理部からプロセス間通信要求を伝えられた場合は、前記通信管理テーブルを検索し、第１のプログラムがプロセス間通信可能な状態にあり、且つ該第１のプログラムとプロセス間通信処理中の第２のプログラムが存在するか否かを判定することを特徴とする情報処理システム。
【請求項３４】請求項３３記載の情報処理システムにおいて、
前記アクセス制御処理部は、現在プロセス間通信可能な状態にある第１のプログラムを特定する情報として、該第１のプログラムを一意に識別可能な情報と、サービスの種類を識別するためのサービス番号と、該第１のプログラムが指定した暗証情報とを前記通信管理テーブルに登録し、
該第１のプログラムとプロセス間通信処理中の第２のプログラムを特定する情報として、該第２のプログラムを一意に識別可能な情報を前記通信管理テーブルに登録して保持することを特徴とする情報処理システム。
【請求項３５】請求項３４記載の情報処理システムにおいて、
前記第２のＯＳは、さらに、プロセス間通信要求内容を登録する通信ログファイルを備え、
前記アクセス制御処理部は、前記第１の監視処理部から伝えられる，前記第１のプログラムによるプロセス間通信要求を前記通信制御ポリシーファイルの記述と照合し、該プロセス間通信要求元が前記通信制御ポリシーに登録されていれば、該第１のプログラムを特定する情報を、前記通信管理テーブルに登録する処理部と、
該プロセス間通信要求元が前記通信制御ポリシーに登録されていなければ、該プロセス間通信要求内容を、前記通信ログファイルに登録する処理部と、
前記第２の監視処理部から伝えられる，前記第３のプログラムによるプロセス間通信要求を前記通信制御ポリシーファイルの記述と照合し、該プロセス間通信要求元が前記通信制御ポリシーにて許可されており、、かつ前記第１のプログラムと通信可能な状態であれば、該第２のプログラムを特定する情報を前記通信管理テーブルに登録して、前記暗証情報を通信管理テーブルから取り出して前記第２の監視処理部に返信する処理部と、
前記第３のプログラムによる該プロセス間通信要求元が前記通信制御ポリシーに登録されていなければ、該プロセス間通信要求内容を、前記通信ログファイルに登録する処理部と、を備えることを特徴とする情報処理システム。
【請求項３６】請求項３５記載の情報処理システムであって、
前記アクセス制御処理部は、さらに、
前記第１の監視処理部または第２の監視処理部からプロセス間通信の終了要求を伝えられた場合には、前記通信管理テーブルから該当する情報を削除する処理部を備えることを特徴とする情報処理システム。
【請求項３７】請求項３６記載の情報処理システムであって、
前記第１の監視処理部は、
前記第１のプログラムからプロセス間通信要求が発行された場合に、該第１のプログラムを特定する情報を取得してから、前記発行されたプロセス間通信要求を前記アクセス制御処理部に伝える情報処理システム。
【請求項３８】請求項３７記載の情報処理システムであって、
前記第２の監視処理部は、
前記第２のプログラムからプロセス間通信要求が発行された場合に、該第２のプログラムを特定する情報を取得してから、前記プロセス間通信の開始要求を前記アクセス制御処理部に伝え、該アクセス制御処理部にて許可される場合には、前記プロセス間通信の開始を許可して、前記第１のプログラムに対して前記暗証情報を伝える処理部と、前記アクセス制御処理部にて許可されない場合には前記プロセス間通信の開始を無効にする処理部とを備えることを特徴とする情報処理システム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、情報処理装置が管理する情報を不正なアクセスから保護する場合に好適なアクセス制御技術に関する。
【０００２】
【従来の技術】一般のコンピュータシステムでは、マルチユーザー・マルチタスクＯＳが備えるユーザー認証機構と、該認証結果に基づいたファイルアクセス制御機構を用いて機密情報ファイルの保護を実現しているケースが多い。具体的には、前記ＯＳが実装された情報処理装置を利用する際に、ユーザーは必ず自己のユーザーＩＤとパスワードを入力し、認証を受ける。前記情報処理装置が管理する全てのファイル各々には、ファイルリードやライト等のアクセスタイプ毎に、ユーザーＩＤとグループＩＤを用いてアクセス可能なユーザーを定義したアクセスコントロールリストがセキュリティ属性情報として割り当てられている。ユーザーがアプリケーションプログラムを介してファイルへアクセスした場合、上記ＯＳは、アクセス要求元となるユーザーのＩＤ及び該ユーザーが所属するグループＩＤを、アクセス対象となるファイルに割り当てられたアクセスコントロールリストと照合し、当該リストに前記ユーザーが含まれている場合に限りアクセスを許可するといった制御を行なう。
【０００３】一方で、インターネットを介した情報発信、情報収集や、各種サービス提供を行なう手段としてＷｏｒｌｄＷｉｄｅＷｅｂ（WWW）のサービスが広く

使われている。WWWでは、ＨｙｐｅｒＴｅｘｔＴｒａｎｓｆｅｒＰｒｏｔｏｃｏｌ（ＨＴＴＰ）と呼ばれる通信プロトコルが、リクエストデータとレスポンスデータの転送に用いられている。また、WWWシステムではコンテンツの不正な差し替えや、ネットワークを経由した機密情報流出を未然に防ぐためのセキュリティ技術がいくつか用意されている。

【０００４】ＨＴＴＰが有するセキュリティ機構として、基本認証（ＢａｓｉｃＡｕｔｈｅｎｔｉｃａｔｉｏｎ）と呼ばれるものがある。基本認証では、認証情報としてユーザーＩＤとパスワードを予めWWWサーバに登録しておき、ブラウザを通じてユーザーから送信されるユーザーＩＤとパスワードを、前記認証情報と比較して認証を行なう。各コンテンツへのアクセス権限設定を記述したポリシーファイルと、該ポリシーに基づくアクセス制御機構も前記WWWサーバに実装されている。同様な機構を、ＣｏｍｍｏｎＧａｔｅｗａｙＩｎｔｅｒｆａｃｅ（ＣＧＩ）プログラムに実装し、ユーザー認証とコンテンツへのアクセス制御を実現することも可能である。

【０００５】WWWでは、他のセキュリティ技術として、認証局から発行される証明書に基づき、ユーザーとWWWサーバの相互認証と通信データの暗号化を行なうことが可能である。この技術は、消費者のクレジットカード番号がネットワーク上に流れる一部のインターネットショッピングのようなサービスにおいて、必須の技術とされている。これらの技術によるユーザー（クライアント）認証は、WWWサーバ側のＯＳが具備するユーザー認証機構とは通常独立したものであるが、前記認証に用いるユーザーＩＤや、ユーザーの証明書を、ＯＳが管理するユーザーアカウントに関連付けて使用することができるWWWサーバプログラムもある。つまり、認証されたユーザーは、ＯＳのアクセス制御下でWWWコンテンツにアクセスする。これらのセキュリティ技術については、“Web Security:A Step-by-Step Reference Guide” Lincoln D. Stein著、Addison-Wesley Pub Co; (ISBN:0201634899) などに記載されている。

【０００６】近年増加しているインターネットサービスプログラム等に潜むセキュリティホールやバグを利用した不正アクセスを解決する他の技術としては、常駐型のファイル監視プログラムを情報処理装置上に設け、定期的にファイルの破壊の有無をチェックする方法が、特開平１０－０６９４１７号公報に開示されている。これと同様の技術として、特定のファイルについてそのサイズの増減やタイムスタンプを定期的にチェックするプログラムを情報処理装置上に設け、該ファイルの変化を検知するツールが、シェアウエアやフリーウエアとして幾つか公開されている。

【０００７】

【発明が解決しようとする課題】先に述べたセキュリティホールやバグに関する情報は、ｈｔｔｐ：／／ｗｗｗ．ｃｅｒｔ．ｏｒｇ／等で報告され、バグ修正用のプログラムも各メーカーから配布されるようになっている。しかし、攻撃者あるいは侵入者は、あらゆる手段を用いて外部ネットワークから情報処理装置への侵入を試みる。たとえば、前記マルチユーザー・マルチタスクなＯＳが具備するアクセス制御機構では、アクセス要求元のユーザーＩＤとその所属グループに基づいてアクセスの可否を判断しているため、ＯＳの管理者のような強い権限をもつユーザーに成りすまして侵入すれば、システム中のいかなるファイルにもアクセスできるので、システム中の情報を書き換えたり、機密情報を盗み出すことがてきてしまう。

【０００８】また、前記ファイルの変化を定期的にチェックする手法では、ファイルが改ざんされた後の検出になるため、不正アクセスが発生したことは分かるが、それを未然に防ぐことができないと共に、ファイルの不正な読み出しについては事後検出もできない。

【０００９】また、前記WWWサーバやＣＧＩプログラムを利用してコンテンツのアクセス制御機能を実現したシステムにおいては、各コンテンツへのアクセス権限設定を記述したポリシーファイルも、その他のファイルと同じく不正アクセスの対象になる。したがって、該ポリシーファイルが破壊あるいは改ざんされた場合には、前記アクセス制御機能が正常に働かない。

【００１０】本発明の目的は、アクセス要求元（サブジェクト）がいかなる権限をもつ場合でも、そのアクセス手段およびアクセス対象（オブジェクト）を制限することが可能なアクセス制御システム及びその方法を提供することにある。

【００１１】本発明の他の目的は、不正なファイルアクセスを未然に防ぐことが可能なアクセス制御システム及びその方法を提供することにある。

【００１２】本発明の他の目的は、アクセス権限設定を記述したポリシーファイルと、該ポリシーファイルに基づいてアクセス制御を施行するプログラムを、外部からの不正アクセスや攻撃から保護可能なアクセス制御システム及びその方法を提供することにある。

【００１３】本発明の他の目的は、例えば機密情報にアクセスしながらサービスを提供するようなアプリケーションプログラムを、外部からの不正アクセスから保護可能なアクセス制御システム及びその方法を提供することにある。

【００１４】

【課題を解決するための手段】上記目的を達成するために、本発明では、ファイルアクセスに関するアクセス制御ポリシーとして、より厳密にアクセス要求を特定する情報、すなわち、アクセス要求元とアクセス実行手段とアクセスタイプとを特定する情報を用いる。

【００１５】さらに具体的には、特定のファイルへのア

クセスを、特定のユーザーが特定のプログラムを用いた場合のみ許可するアクセス制御ポリシーを記述したポリシーファイルを作り、アクセス制御手段が、ファイルへのアクセスが発生したときに、前記ポリシーファイルの記述に従ってアクセスの正当性すなわち可否を判定する。前記アクセス制御ポリシーには、アクセス対象となるファイルの名称と、アクセスが許可されたユーザー名とプログラム名の組合せを、ファイルのオープン・リード・ライト・削除・リネームといったアクセスタイプ毎に予め規定しておく。

【００１６】また、不正なファイルアクセスを未然に防ぐために、本発明では、前記ファイルアクセスを監視するファイルＩ／Ｏ（Ｉｎｐｕｔ／Ｏｕｔｐｕｔ）フック手段を前記情報処理装置に設け、該ファイルＩ／Ｏフック手段は、前記アクセスを検知した時に、当該アクセスタイプ及びアクセス対象となるファイルの名称と、該アクセスの要求元となるユーザー名及びプログラム名を取得して前記アクセス制御手段に送信する。前記アクセス制御手段は、受信内容を前記ポリシーファイルの内容と照合し、該ポリシーに違反するアクセスであれば当該アクセスを無効にし、前記ファイルＩ／Ｏフック手段を経由して前記アクセス要求元にエラーを返す。

【００１７】また、前記ポリシーファイルと前記アクセス制御手段を保護するために、本発明では、１台の情報処理装置上に２つのＯＳと、両ＯＳ間でデータ通信するためのプロセス間通信手段と、両ＯＳが互いに排他的に占有するメモリや磁気ディスクおよびネットワークデバイスを設ける。前記２つのＯＳのうち一方をアクセス監視対象となるサービス用ＯＳとして使用し前記ファイルＩ／Ｏフック手段を前記サービス用ＯＳのカーネルレベルのモジュールとして設ける。もう一方をセキュリティ用ＯＳとして使用し、前記アクセス制御手段および前記ポリシーファイルを占有する磁気ディスクに格納すると共に、前記アクセス制御手段をプロセスとして動作させ、該アクセス制御手段とポリシーファイルを前記サービス用ＯＳ上の、サービスを受けるユーザが使うプロセスからアクセスできないようにする。

【００１８】一方、よりセキュリティを高めるために、機密情報を取り扱うようなアプリケーションプログラム（以後、機密プログラムと称す）とのプロセス間通信に関するアクセス制御ポリシーとして、より厳密にアクセス要求を特定する情報、すなわち、アクセス要求元とアクセス実行手段とを特定する情報を用いてもよい。具体的には、特定の機密プログラムとのプロセス間通信を、特定のユーザーが特定のプログラムを用いた場合のみ許可するアクセス制御ポリシーを記述したポリシーファイルを作り、複数のアプリケーションプログラム間で通信が発生したときに、プロセス間通信手段が当該通信を検知してアクセス制御手段に通知し、アクセス制御手段が前記ポリシーファイルの記述に従って通信の正当性すなわち可否を判定する。前記アクセス制御ポリシーには、通信機能を備える機密プログラムの名称と、当該機密プログラムとの通信が許可されたユーザー名とプログラム名の組合せを予め規定しておく。

【００１９】この場合は，前記機密プログラムと当該機密プログラムがアクセスする機密情報ファイルについても、前記セキュリティ用ＯＳが占有する磁気ディスクに格納すると共に、前記機密プログラムをセキュリティ用ＯＳ上のプロセスとして動作させることでサービス用ＯＳ側からの不正アクセスから保護する。もしくは、セキュリティ用ＯＳやサービス用ＯＳとは別の第３のＯＳと、各ＯＳ間で通信するためのプロセス間通信手段と、各ＯＳが互いに排他的に占有するメモリや磁気ディスクおよびネットワークデバイスを設け、前記機密プログラムと当該プログラムがアクセスする機密情報ファイルを、前記第３のＯＳが占有する磁気ディスクに格納すると共に、第３のＯＳ上のプロセスとして動作させることで、サービス用ＯＳ側からの不正アクセスから保護する。

【００２０】本発明におけるＯＳとは、ユーザーまたはプログラムからの要求に応じて記憶媒体中のデータ、ファイルへのアクセスを実行する機能と、アクセス要求元のユーザーやプログラムを識別する機能と、排他制御機構により占有する記憶手段を有するプログラムモジュールを意味しており、

・データ（ファイル）アクセスを管理しており、検知が可能である。

・アクセス要求元のユーザーを識別できる。

・アクセス要求元のアプリケーションを識別できる。

【００２１】といった特徴を持つ。したがって、一般的にＯＳと呼ばれるものに限らず上記特徴を持つものであれば、本発明を適用することは可能である。

【００２２】また，サービス提供とは，リクエストに応じて所定の処理を実行し、その処理結果を要求元に返すことを指し，また，サービス利用とは，上記サービス提供をおこなうものに対して、リクエストを発行し、その処理結果を受け取ることを指す。

【００２３】また，機密情報とは、システム利用者のうち、権限を持つものだけがその内容および存在自体を知り得るような情報を指し，機密プログラムとは、このような情報を扱いながら所定を処理を実行するプログラムを指す。

【００２４】本発明を実現するのに必要な各プログラム（コード、モジュール、ユニットともいう）は，ネットワークに接続される他のサーバからコンピュータが読み取り可能な媒体，すなわちネットワーク上の伝送信号，またはＣＤＲＯＭ，ＦＤなどの可搬型記憶媒体，を経由して，事前にまたは必要なときに導入してもよい。

【００２５】

【発明の実施の形態】以下、図を用いて本発明の実施の

一形態を説明する。

【００２６】図１は、本発明のアクセス制御システムの一構成例である。１００はサーバ情報処理装置であり、サービス用オペレーティングシステム（ＯＳ）１０３が管理するメモリ１０１、当該サービス用ＯＳ１０３が占有するディスクコントローラ１１２ａと磁気ディスク１１４ａとＬＡＮコントローラ１１３、セキュリティ用ＯＳ１０４が管理するメモリ１０２と、当該セキュリティ用ＯＳ１０４が占有するディスクコントローラ１１２ｂと磁気ディスク１１４ｂその他を備える。

【００２７】複数ＯＳ制御プログラム１１６は、サーバ情報処理装置上で複数のＯＳ（本実施例ではサービス用ＯＳ１０３とセキュリティ用ＯＳ１０４）が動作するための各種の制御をする部分で、各ハードウェアの初期化および、メモリと磁気ディスクの分割占有処理、ＣＰＵのスケジューリング、割り込み処理等を行う。ＯＳ間通信処理部１０８は、複数ＯＳ制御プログラム１１６の中に存在し、サーバ情報処理装置上で動作するＯＳ間での通信機能を提供する。やサーバ情報処理装置１００の起動時には、前記複数ＯＳ制御プログラム１１６が各ハードウェアの初期化およびＯＳ毎のハードウェア分割占有処理等を行い、サービス用ＯＳ１０３とサーバプログラム１０９がディスクコントローラ１１２ａを介して磁気ディスク１１４ａからメモリ１０１上にロードされると共に、セキュリティ用ＯＳ１０４とアクセス制御プログラム１１０がディスクコントローラ１１２ｂを介して磁気ディスク１１４ｂからメモリ１０２上にロードされる。前記ＯＳ間通信処理部１０８は、ファイルＩ／Ｏフックプログラムと前記アクセス制御プログラム１１０との間でデータ交換を行なうために使用するものとし、前記サーバプログラム１０９からは直接使用できないインタフェースとなっている。

【００２８】なお、ＯＳ間通信処理部１０８は、サービス用ＯＳ１０３に含まれるプロセス間通信プログラムとして、構成することも可能である。

【００２９】また、サービス用ＯＳ１０３は、流通している既存のマルチユーザー・マルチタスクなＯＳと同様、ユーザーの識別・認証機能を具備しているものとする。前記サーバ情報処理装置１００と複数のクライアント情報処理装置１２０は、ローカルエリアネットワーク（ＬＡＮ）１１５を介して相互に接続されており、前記サーバプログラム１０９は、ＬＡＮコントローラ１１３とＬＡＮ１１５を介して、クライアントプログラム１２１からのリクエストの受信と、当該クライアントプログラム１２１へのレスポンスの送信を行なう。サーバプログラム１０９とクライアントプログラム１２１は、例えばWWWサーバとブラウザに相当するプログラムである。なお、前記サーバ情報処理装置１００とクライアント情報処理装置１２０は、電話回線やインターネットを介して接続されていてもよい。また、サービス用ＯＳ１０３とセキュリティＯＳとは、ＯＳ間通信のインタフェースを非公開にすれば別々の装置上にあっても良く、セキュリティ的にも問題はない。

【００３０】前記サーバプログラム１０９のようにＯＳ上で動作するアプリケーションプログラムは、一般にユーザーレベルのプロセスと呼ばれている。該サーバプログラム１０９は、前述のようにクライアントプログラム１２１からのリクエストに基いて処理を実行するものであり、言い換えればネットワークからの攻撃の対象にもなり得るプログラムである。これに対し、ファイルＩ／Ｏフックプログラム１０６やファイルシステムドライバ１０７のようにＯＳの一機能として動作するプログラムは、一般にカーネルレベルのモジュールと呼ばれており、多くのコンピュータシステムではアクセス制御機能をカーネルレベルのモジュールとして実装している。

【００３１】図１のサーバ情報処理装置１００のように、１台の情報処理装置上に複数のＯＳを同時に動作させるための技術としては、仮想計算機あるいはマイクロカーネルがある。その他、リアルタイムＯＳを一般的なＯＳのカーネルレベルのモジュールとして実装すると共に、前記リアルタイムＯＳがシステム障害を検知すると、リアルタイム処理を継続しながらもシステムを自動的に再起動する方法が、特開平１１－０２４９４３号公報に開示されている。また、ＯＳ間通信処理部１０８のような、異種ＯＳ上のプロセス間通信を実現する方法は、特開平１１－０８５５４６号公報に開示されている。これらを本発明の前提条件として、引続き実施の形態を述べる。

【００３２】図２は、前記ポリシーファイル２００のデータ構造を示したものである。

【００３３】２１０から２１２は、ポリシー記述の一例を示したものである。オブジェクト名２０１は、アクセスを制限すべきファイルの名称を示す。オブジェクト名として、２１０や２１２のようにファイル単位の指定と、２１１のようなディレクトリ単位の指定が可能となっている。ディレクトリ単位の指定では、該ディレクトリ名に続くファイル名を所定の文字、記号（たとえばアスタリスク（＊））で表記する。前記オブジェクトに対して禁止すべきアクセスを、禁止されたアクセスのタイプ２０２に示す。エラーコード２０３は、前記禁止されたアクセスのタイプ２０２が発生した際に、当該アクセス発行元（サブジェクト）となるプログラムに返すべきエラーコードを示す。例外サブジェクト２０４は、特別にアクセスを許されたプログラムの名称である。プログラムのハッシュ値２０５は、前記例外サブジェクトとして指定されたプログラムファイルの特徴値（ハッシュ値）をたとえば８バイトで表したものである。ユーザー名２０６は、前記例外サブジェクト２０４のプログラムを利用可能なユーザーを、前記サービス用ＯＳ１０３が管理するユーザー名またはグループ名で表したものであ

る。

【００３４】つまり前記ポリシーは、前記プログラムのハッシュ値２０５に登録されたハッシュ値を有するプログラムが前記例外サブジェクトとして登録され、且つ該プログラムをユーザー名２０６に登録されたユーザーあるいはグループのメンバーが利用している場合に限り、例外としてアクセスを許可することを表している。同時に、これら条件が整っていない場合は、アクセス発行元のプログラムに対して前記エラーコード２０３を返すことを意味する。

【００３５】前記ポリシーファイル２００の設定は、システムのセキュリティ管理者が行なうものとする。例えば、ＨＴＭＬファイルなどのWWWコンテンツを侵入者によって不正に書き換えられないようにするには、該ＨＴＭＬファイルへのライトアクセスを原則として禁止しておき、例外としてコンテンツ管理者に任命されているユーザーが特定のＨＴＭＬ編集用プログラムを用いた場合に限りライトアクセスを許可するといったポリシーを前記ポリシーファイル２００に記述すればよい。

【００３６】例外サブジェクトには、サーバプログラム１０９のように、ネットワークからの攻撃を受けやすいプログラムを指定しないことが重要である。更に例外サブジェクトとして登録するプログラムを、ＣＤ－ＲＯＭ等の取り外し可能な記憶媒体に格納しておき、必要なときだけ媒体を装着して使用すればより確実に保護できるので、さらなるセキュリティ効果が期待できる。図１の構成に当該媒体の駆動装置を追加し、ディスクコントローラ１１２ａが当該駆動装置に対応すれば、このような記憶媒体が使用可能となる。

【００３７】図３は、アクセスログファイル３００のデータ構造の一例を示したものである。アクセスログファイル３００は、前記ポリシーファイル２００にて禁じられているアクセスが発生した事実を、アクセス制御プログラム１１０のアクセスログ登録ルーチン４０６が書き込むためのファイルであり、当該アクセスが発生した日時３０１と、当該アクセスの対象となったファイルを表すオブジェクト名３０２と、当該アクセスのタイプ３０３、当該アクセスを発行したプログラムを表すサブジェクト名３０４、そして前記プログラムを利用していたユーザーを表すユーザー名３０５から構成される。

【００３８】図４に、ファイルＩ／Ｏフックプログラム１０６と、アクセス制御プログラム１１０の構成を示す。

【００３９】ファイルＩ／Ｏフックプログラム１０６は、前記サーバ情報処理装置１００の起動時に、サービス用ＯＳ１０３と共にメモリ１０１上にロードされる。アクセス制御プログラム１１０は、前記サーバ情報処理装置１００の起動時に、セキュリティ用ＯＳ１０４と共にメモリ１０２上にロードされる。

【００４０】図５と図６を用いて、本発明のアクセス制御の概要を記す。

【００４１】図５は、Ｉ／Ｏマネージャ１０５と、ファイルＩ／Ｏフックプログラム１０６と、アクセス制御プログラム１１０の三者間における処理の流れを示すものである。５０１は、サーバプログラム１０９が発行したファイルアクセスが、Ｉ／Ｏマネージャ１０５を経由してファイルＩ／Ｏフックプログラム１０６に到達するまでの通信経路であり、サービス用ＯＳ１０３の上で動作する全てのアプリケーションプログラムからのファイルアクセスについて共通なものである。ファイルＩ／Ｏフックプログラム１０６には、図６に示すＩ／Ｏパケット６００のデータが渡される。Ｉ／Ｏパケット６００は、Ｉ／Ｏフックプログラム１０６に渡される前にサービス用ＯＳ１０３が具備するアクセス制御機構をパスしていることと仮定して説明するが、これに限ったものではなく、前記アクセス制御プログラム１１０のチェックをパスしてからサービス用ＯＳ１０３が具備するアクセス制御機構のチェックを受けるものであってもよい。

【００４２】図６は、前記三者間を流れるパケットのデータ構造を示したものである。６０１はアクセス対象のファイルを表すオブジェクト名である。６０２は前記オブジェクトへのアクセスタイプを示す。６０３は、当該アクセスを発行したプログラムのプロセスＩＤを示す。６０４は当該アクセスの処理結果を示すステータスを示す。６０５は当該アクセスに関連するデータの長さを示すものであり、例えば前記アクセスタイプ６０２がファイルへの書込み（ライト）要求であれば、書込みデータがデータ領域６０７に格納され、当該書込みデータの長さがデータ長６０５に格納される。６０６は、前記データ領域６０７へのポインタである。

【００４３】図５において、５０２は、ファイルＩ／Ｏフックプログラム１０６が検知したファイルオープン要求に関するアクセス権限チェックと、アクセスログ３００への書き込みを、アクセス制御プログラム１１０に対してリクエストするに用いる通信経路である。前記リクエストのデータ構造を、図６のリクエストパケット６１０に示す。サブジェクト名６１１は、ファイルアクセスの発行元となるプログラムの名称であり、ユーザー名６１２は、前記プログラムの利用者をユーザー名とグループ名で表すものである。サブジェクト名６１１及びユーザー名６１２は、受信したＩ／Ｏパケット６００に含まれるプロセスＩＤ６０３を指定して、前記サービス用ＯＳ１０３として用いる標準のＯＳに対してシステムコールを発行することで取得できる。

【００４４】前記リクエストパケット６１０は、ＯＳ間通信処理部１０８が当該パケットのデータを、ファイルＩ／Ｏフックプログラムのメモリ空間６３０からアクセス制御プログラムのメモリ空間６４０に複写することで伝達される。アクセス制御プログラム１１０では、受信したリクエストパケットに含まれる情報を、ポリシーフ

ファイル２００の内容と照合し、その結果をレスポンスパケット６２０に登録して前記ファイルＩ／Ｏフックプログラムに伝達する。この伝達は、前記通信経路５０２と同様に、ＯＳ間通信処理部１０８が前記パケットのデータを、アクセス制御プログラムのメモリ空間６４０からファイルＩ／Ｏフックプログラムのメモリ空間６３０に複写することで達成される。５０３は、当該レスポンスパケットの通信経路を示すものである。

【００４５】ファイルＩ／Ｏフックプログラム１０６は、受信したレスポンスパケット６２０の内容から当該アクセスの可否を判断し、アクセス不可であればエラーコード２０３をＩ／Ｏパケット６００のステータス６０４に設定し、当該Ｉ／Ｏパケットを前記Ｉ／Ｏマネージャ１０５に返す。前記ステータス６０４にエラーコードが設定されている場合、前記Ｉ／Ｏマネージャ１０５は、通信経路５０６を用いてファイルアクセス発行元のサーバプログラム１０９に当該エラーを返す。一方、前記ステータス６０４にエラーコードが設定されていなければ、通信経路５０５を用いてファイルシステムドライバ１０７及びディスクコントローラ１１２ａを介して磁気ディスク１１４ａへのファイルアクセスを続行し、通信経路５０６を用いて当該アクセスの結果を前記サーバプログラム１０９に返す。

【００４６】本発明のアクセス制御の具体的処理内容を、図７から図１６を用いて説明する。

【００４７】図７は、ファイルＩ／Ｏフックプログラム１０６が具備するファイルＩ／Ｏフックルーチンの処理内容を示すフロー図である。

【００４８】サーバプログラム１０９が、ステップ７０１でクライアントプログラム１２１からのリクエストを受信し、ステップ７０２で前記リクエストに応じたファイルアクセスをサービス用ＯＳ１０３に対して発行した場合を想定する。

【００４９】当該ファイルアクセスは、通信経路５０１を経由して、ファイルＩ／Ｏフックルーチン４００にＩ／Ｏパケット６００として渡される（ステップ７０３）。ステップ７０３からステップ７１２は、ファイルＩ／Ｏフックルーチン４００の処理フローである。ステップ７０４では、Ｉ／Ｏパケット６００に含まれるプロセスＩＤ６０３から、ファイルアクセス発行元のサブジェクト名６１１とユーザー名６１２を取得する。

【００５０】ステップ７０５からステップ７０８では、アクセスタイプ６０２を調べ、ファイルオープン、ファイルクローズ、ファイルリード又はライト、またはファイル削除又はリネームにそれぞれ対応する処理ルーチンを実行する。これらに該当しない場合、又は対応する処理ルーチンを実行した後は、通信経路５０４を経由して、ステップ７１３にてＩ／Ｏマネージャ１０５の処理に戻る。

【００５１】図８を用いてオープン処理ルーチン４０１の処理フローを説明する。ステップ８０１では、アクセス日時として現在の日付と時刻を取得する。ステップ８０２では、リクエストパケット６１０を作成すると共に、当該パケット中のエラーコード２０３とハッシュ値２０５を０に初期設定する。次にステップ８０３にて、アクセス制御プログラム１１０のオープン制御ルーチン４０５をコールすると共に、前記リクエストパケット６１０をオープン制御ルーチン４０５に渡す。

【００５２】オープン制御ルーチン４０５の処理フローを、図９を用いて説明する。ステップ９０１では、受信したリクエストパケット６１０のうち、オブジェクト名６０１とアクセスタイプ６０２の内容を、ポリシーファイル２００と照合する。ステップ９０２では、当該アクセスが禁止されたアクセスのタイプとして登録されているかどうかを判別する。このとき、禁止されたアクセスタイプに該当しなければ正当なアクセスとみなし、ステップ９０３にて、許可されたアクセスタイプをレスポンスパケット６２０の認可アクセス６１３に登録し、オープン処理ルーチン４０１の処理に戻る。ここで、許可されたアクセスタイプとは、オブジェクト名６０１で示されるファイルに対するリードとライトの内、サブジェクト名６１１で示されるプログラムから実行可能なアクセスタイプのことを表す。したがって、オブジェクト名６０１で示されるファイルに対して、ポリシーファイル２００の中でリードとライト共に許可されていれば、認可アクセス６１３にはリードとライトの両方を登録する。また、リードのみが許可されている場合はリードのみを、ライトのみが許可されている場合はライトのみを前記認可アクセス６１３に登録する。リードとライトの両方を禁じる場合には、ポリシーとして当該ファイルのオープンを禁止するよう、予めポリシーファイル２００に記述しておけばよい。

【００５３】ステップ９０２にて禁止されたアクセスだと判断した場合、ステップ９０４にて、前記サブジェクト名６１１とユーザー名６１２が、共にポリシーファイル２００に記述された例外サブジェクト２０４とユーザー名２０６に該当するか否かを判別する。このとき、サブジェクト名については該プログラムファイルのパス名が完全に一致するか否かを判別する。またユーザー名については、ユーザー名６１２に含まれるユーザー名とグループ名のいずれかが一致するか否かを判別する。判別の結果、例外サブジェクト２０４かつユーザー名２０６に該当する場合、ステップ９０５にて当該プログラムファイルのハッシュ値２０５をポリシーファイル２００から取得して、レスポンスパケット６２０に設定する。また、ステップ９０６では、ステップ９０３と同様に、許可されたアクセスタイプをレスポンスパケット６２０の認可アクセス６１３として設定する。その後、ステップ９０８で、該当するエラーコード２０３をレスポンスパケット６２０に設定してから、オープン処理ルーチン４

０１へ戻る。ここでのエラーコード設定は、オープン処理ルーチン４０１のステップ８０４からステップ８１２の処理の中で意味を持つものである。

【００５４】ステップ９０４にて、例外サブジェクト２０４とユーザー名２０６とに該当しないと判断した場合、ステップ９０７にて、当該アクセスの内容をアクセスログファイル３００に書込む。その後、ステップ９０８で、該当するエラーコード２０３をレスポンスパケット６２０に設定してから、オープン処理ルーチン４０１へ戻る。

【００５５】次に、図８において、ステップ８０４ではレスポンスパケット中のハッシュ値２０５の値を調べ、０であれば例外として認められたアクセスではないと判断し、ステップ８０５でエラーコードの値をチェックする。エラーコードが０でなければ禁止されたアクセスであったと判断し、ステップ８０６にて、Ｉ／Ｏパケット６００のステータス６０４に、当該エラーコード２０３を設定して処理を終了する。ステップ８０５で、エラーコードが０であれば、正当なアクセスであったと判断し、ステップ８１０にて当該アクセス情報を後述するオープンファイルテーブルの先頭アドレスに登録してから処理を終了する。

【００５６】ステップ８０４でハッシュ値が０でない場合は、例外として認められたサブジェクトであると判断し、ステップ８０７にてサブジェクト名６０１で示されるプログラムファイルのハッシュ値を算出し、ステップ８０８にてレスポンスパケット中のハッシュ値２０５と比較する。両者が等しければ、例外サブジェクト２０４に相当するとみなして、ステップ８１２にて当該アクセス情報をオープンファイルテーブルの先頭アドレスに登録してから本ルーチンの処理を終了する。ハッシュ値が等しくなければ、前記プログラムファイルが不正なプログラムであるとみなして、ステップ８０９にて前記リクエストパケットにおけるサブジェクト名６１１を例外サブジェクトにならないように、例えばサブジェクト名としてnullデータに、変更し、、ステップ８１０にてアクセスログ登録ルーチン４０６をコールすると共に前記リクエストパケット６１０をアクセスログ登録ルーチン４０６に渡す。

【００５７】該アクセスログ登録ルーチン４０６の処理フローを、図１４を用いて説明する。ステップ１４０１で、リクエストパケット６１０の内容をポリシーファイル２００と照合する。予め前記ステップ８０９で例外サブジェクト扱いにならぬようにサブジェクト名６１１をクリアしておくことにより、必ず禁止されたオープンアクセスとして扱われる。ステップ１４０２にて、該当するエラーコードをポリシーファイル２００から読み出してレスポンスパケット６２０に設定し、当該パケットを返す。ステップ１４０３にて当該アクセス内容を、アクセスログファイル３００に書き込み、オープン処理ルーチン４０１の処理に戻る。

【００５８】図８のオープン処理ルーチン４０１において、ステップ８１１でレスポンスパケット中のエラーコード２０３をＩ／Ｏパケットのステータス６０４に設定して本ルーチンの処理を終了する。

【００５９】図１０を用いてオープンファイルテーブルについて説明する。オープンファイルテーブル１０００は、現在オープン中のファイルに関する情報を格納した構造体データ１００２の集合である。１個の構造体には１件のオープンファイルの情報を記憶しており、ファイルＩ／Ｏフックプログラム１０６では、各構造体を先頭アドレス１００１とポインタ１００３によりリストとして管理する。本オープンファイルテーブル１０００の一例を図１１に示す。

【００６０】オープンファイルテーブル１０００に登録された情報に該当するアクセスであれば、該アクセスを許可することになるため、オープンファイルテーブル１０００を不正に書換えられないよう保護することが重要である。サービス用ＯＳ１０３として、プロセス毎に独立したメモリ空間が割当てられると共にメモリ空間の排他制御機構が働くＯＳを使うことで、前記オープンファイルテーブル１０００も別プロセスからは不正に書換えできない仕組みにすることが可能になる。

【００６１】図１２は、クローズ処理ルーチン４０２の処理フローを示したものである。ステップ１２０１では、受信したＩ／Ｏパケット６００の中にあるオブジェクト名６０１と、プロセスＩＤ６０３、並びにプロセスＩＤ６０３から取得したサブジェクト名６１１およびユーザー名６１２の組み合わせと同じものをオープンファイルテーブル１０００から検索する。ステップ１２０２で、該当する情報がテーブルにあれば、ステップ１２０３にて該当する情報をテーブルから削除する。一方、テーブルに存在しなければ、ポリシーファイル２００に登録されていないファイルへのクローズ要求とみなし、本ルーチンの処理を終了する。

【００６２】図１３は、リード・ライト処理ルーチン４０３の処理フローを表したものである。ステップ１３０１では、受信したＩ／Ｏパケット６００の中にあるオブジェクト名６０１と、プロセスＩＤ６０３と、プロセスＩＤ６０３から取得したサブジェクト名６１１と、ユーザー名６１２との組み合わせと同じものをオープンファイルテーブル１０００から検索し、更にアクセスタイプ６０２が認可されたアクセスタイプ６１３に含まれるか否かをチェックする。ステップ１３０２にて、前記オブジェクト名６０１と、プロセスＩＤ６０３と、サブジェクト名６１１と、ユーザー名６１２と、アクセスタイプ６０２との組合せがテーブル１０００にあれば、本ルーチンの処理を終了する。このことは、オープンファイルテーブル１０００に登録されたアクセスであれば正当なアクセスであるとみなし、アクセス制御プログラム１１

０の処理を行わないことを意味する。これは、本発明のアクセス制御によって生じるシステムのパフォーマンス低下を軽減する効果がある。

【００６３】ステップ１３０２にて、オープンファイルテーブルに登録されたアクセスでなければ前記ポリシーファイル２００により禁止されたアクセスとみなし、ステップ１３０３でアクセス日時を取得し、ステップ１３０４でリクエストパケット６１０を作成する。このとき、エラーコード２０３を０に初期設定する。この後、ステップ１３０５にてアクセスログ登録ルーチン４０６の処理にジャンプする。

【００６４】図１４のアクセスログ登録ルーチン４０６において、ステップ１４０１ではリクエストパケット６１０の内容をポリシーファイル２００と照合し、ステップ１４０２にて、該当するエラーコードをポリシーファイル２００から読み出してレスポンスパケット６２０に設定する。次に、ステップ１４０３にて当該アクセス内容を、アクセスログファイル３００に書き込み、リード・ライト処理ルーチン４０３の処理に戻る。リード・ライト処理ルーチン４０３では、図１３のステップ１３０６において、レスポンスパケットで受信したエラーコードをＩ／Ｏパケットのステータス６０４に設定し、本ルーチンの処理を終了する。

【００６５】図１５は、削除・リネーム処理ルーチン４０４の処理フローを表したものである。図８のオープン処理ルーチン４０１と同じ処理内容のステップには同じ番号を付している。

【００６６】ステップ１５０３にて、アクセス制御プログラム１１０の削除・リネーム制御ルーチン４０７をコールすると共に、前記リクエストパケット６１０を削除・リネーム制御ルーチンに渡す。

【００６７】削除・リネーム制御ルーチン４０７の処理フローを、図１６を用いて説明する。図９のオープン制御ルーチン４０５と同じ処理内容のステップには同じ番号を付している。ステップ１６０２では、当該アクセスが禁止されたアクセスのタイプとして登録されているかどうかを判別する。禁止されたアクセスタイプに該当しなければ正当なアクセスとみなし、削除・リネーム処理ルーチン４０４の処理に戻る。

【００６８】ステップ１６０２にて禁止されたアクセスだと判断した場合、図９と同様にステップ９０４、９０５の処理を行い、ステップ１６０３にて、例外サブジェクト２０４とユーザー名２０６に該当しないと判断した場合、図９と同様にステップ９０７の処理を行う。

【００６９】ステップ９０５または９０７の処理を済ませると、ステップ９０８で、該当するエラーコード２０３をレスポンスパケット６２０に設定してから、削除・リネーム処理ルーチン４０４へ戻る。

【００７０】次に図１５において、図８と同様にステップ８０４の処理を行う。ステップ１５０５では、エラーコードが０であれば、正当なアクセスであったと判断し、本ルーチンの処理を終了する。ステップ８０７にてサブジェクト名６０１で示されるプログラムファイルのハッシュ値を算出し、ステップ８０８にてレスポンスパケット中のハッシュ値２０５と比較して、両者が等しければ、例外サブジェクト２０４に相当するとみなして本ルーチンの処理を終了する。ハッシュ値が等しくない場合は図８と同様の処理を行い、アクセスログ登録ルーチン４０６を呼び出す。当該ルーチンが処理を済ませると削除・リネーム処理ルーチン４０４の処理に戻る。

【００７１】図１５において、ステップ１５１１でレスポンスパケット中のエラーコード２０３をＩ／Ｏパケットのステータス６０４に設定して本ルーチンの処理を終了する。

【００７２】以上、本発明の実施の一形態を説明したが、ＯＳが備えるファイルシステムの種類によっては、ファイルオープン要求を表すＩ／Ｏパケットの中に、目的となるアクセスタイプを付属情報として含むものがある。つまり、ファイルのリードアクセスをするためのオープン要求なのか、リード・ライトのためのオープン要求なのかを、前記Ｉ／Ｏパケット６００中のアクセスタイプ６０２から検知できる。このようなファイルシステムを前提とした場合には、図７に示すファイルＩ／Ｏフックルーチン４００の処理フローと、図８に示すオープン処理ルーチン４０１の処理フローと、更に図９に示すオープン制御ルーチン４０５の処理フローとを変更することにより、前記オープンファイルテーブル１０００の管理が不要なアクセス制御システムが実現できる。

【００７３】図７においてはファイルＩ／Ｏフックルーチン４００の処理フローから、ステップ７０６と、ステップ７０７と、ステップ７１０と、ステップ７１１を省く。つまり、ファイルオープン要求と、ファイル削除要求と、ファイルのリネーム要求のいずれかを検知し、ファイルのオープン要求であれば、ステップ７０９にてオープン処理ルーチン４０１の処理に移り、削除又はリネーム要求であれば、ステップ７１２にて削除・リネーム処理ルーチン４０４の処理に移るよう、ファイルＩ／Ｏフックルーチン４００の処理フローを変更すればよい。

【００７４】図８においては，オープン処理ルーチンの処理フローから、ステップ８１２の処理を省く。つまり、ステップ８０８でのハッシュ値照合処理の結果、等しければ本ルーチンの処理を終了するよう、オープン処理ルーチン４０１の処理フローを変更する。

【００７５】更に、図９で説明したオープン制御ルーチン４０５の処理フローは、図２６に示す処理フローに変更する。図２６のステップ２８０１における処理，すなわち，受信したリクエストパケット６１０のうち、オブジェクト名６０１とアクセスタイプ６０２の内容を、ポリシーファイル２００と照合するといった処理において、このときアクセスタイプ６０２の中に含まれる前記

オープン要求の付属情報（リードやライト）まで取得して、前記ポリシーファイル２００と照合する。本発明では、前記オープン要求の付属情報として、リードとライトの両方が含まれる場合、リードとライトの両方、もしくはいずれか一方が禁止されたアクセスとしてポリシーファイル２００に記述されていれば、当該オープン要求を禁止されたアクセスとしてみなす。

【００７６】ステップ２８０２では、当該アクセスが禁止されたアクセスのタイプとして登録されているかどうかを判別する。このとき、禁止されたアクセスタイプに該当しなければ正当なアクセスとみなし、ステップ２８０３にて、レスポンスパケット６２０のエラーコード２０３と、ハッシュ値２０５を共に０を設定し、オープン処理ルーチン４０１の処理に戻る。リードとライトの両方を禁じる場合には、ポリシーとして当該ファイルのオープンを禁止するよう、予めポリシーファイル２００に記述しておけばよい。

【００７７】ステップ２８０２にて禁止されたアクセスだと判断した場合、ステップ２８０４にて、前記サブジェクト名６１１とユーザー名６１２が、共にポリシーファイル２００に記述された例外サブジェクト２０４とユーザー名２０６に該当するか否かを判別する。このとき、サブジェクト名については該プログラムファイルのパス名が完全に一致するか否かを判別する。またユーザー名については、ユーザー名６１２に含まれるユーザー名とグループ名のいずれかが一致するか否かを判別する。判別の結果、例外サブジェクト２０４かつユーザー名２０６に該当する場合、ステップ２８０５にて当該プログラムファイルのハッシュ値２０５をポリシーファイル２００から取得して、レスポンスパケット６２０に設定する。その後、ステップ２８０８で、該当するエラーコード２０３をレスポンスパケット６２０に設定してから、オープン処理ルーチン４０１へ戻る。

【００７８】ステップ２８０４にて、例外サブジェクト２０４とユーザー名２０６とに該当しないと判断した場合、ステップ２８０７にて、当該アクセスの内容をアクセスログファイル３００に書込む。その後、ステップ２８０８で、該当するエラーコード２０３をレスポンスパケット６２０に設定してから、オープン処理ルーチン４０１へ戻る。

【００７９】次に、本発明のアクセス制御システムのうち、プロセス間通信に関するアクセス制御について説明する。図１７は、機密性の高い情報と、それにアクセスするプログラムを不正アクセスから保護するためのアクセス制御システムの一構成例であり，図１と同じ構成要素には，同じ番号を付している。この構成は、情報の機密性を確保しつつ、外部ネットワークに対してサービスを提供する場合に有効であり、高い安全性を確保することができる。

【００８０】図１７の構成は，図１の構成と同様であるが，さらに，機密用ＯＳ１２２が管理するメモリ１７０８と、当該機密用ＯＳ１２２が占有するディスクコントローラ１１２ｃと磁気ディスク１１４ｃとＬＡＮコントローラ１２３その他を備える。

【００８１】前記サービス用ＯＳ１０３が占有するＬＡＮコントローラ１１３ａは、インターネット等の外部ネットワーク１７０６に接続されたクライアント情報処理装置１２０等との通信に用いる。一方、前記機密用ＯＳ１２２が占有するＬＡＮコントローラ１１３ｃは、内部ネットワーク１７０７に接続された機器との通信に用いる。

【００８２】複数ＯＳ制御プログラム１１６は、機密用ＯＳ１２２まで含めた複数のＯＳの制御を行う。

【００８３】前記サービス用ＯＳ１０３とセキュリティ用ＯＳ１０４と機密用ＯＳ１２２は、各々ドライバ部１１７とドライバ部１１８とドライバ部１１９を含んでおり、各ＯＳ上のアプリケーションプログラムは、必ずこれらドライバを経由してＯＳ間通信処理部１０８へアクセスする。この方法は、すでに説明したプロセス間通信方法と同様なものである。

【００８４】一般的なプロセス間通信の種類には、共有メモリやセマフォ、メッセージキューなど様々なものがあり、いずれも予め取り決められた名称や番号を用いて通信相手を識別することになっている。

【００８５】更に本発明は、以下の特徴を持つ。すなわち，プロセス間通信の不正利用を防止するために、プロセス間通信に関するセキュリティポリシーを記述した通信制御ポリシー１８００を前記磁気ディスク１１４ｂに格納しておき、前記ＯＳ間通信処理部１０８を利用したプロセス間通信の要求については、セキュリティ用ＯＳ１０４側で動作する通信制御プログラム１７００が、前記通信制御ポリシー１８００の記述に基づいてアクセス制御を行う。また、当該プロセス間通信に関するアクセスログを、前記磁気ディスク１１４ｂ中の通信ログ１９００に記録する機能を備える。また、前記通信制御プログラム１７００は、実行中のプロセス間通信に関する情報を、セキュリティ用ＯＳ１０４が管理するメモリ１０２の中に設けた通信管理テーブル２０００にて保持する。

【００８６】中継プログラム１７０１は、前記サービス用ＯＳ１０３の上で動作するアプリケーションプログラムの１つである。本実施例では、外部ネットワークに接続されたクライアント情報処理装置１２０からの要求に応じて、後述する機密プログラム１７０３とのプロセス間通信を行ない、処理結果を前記クライアント情報処理装置１２０に返信する役割をもったプログラムを中継プログラムと称す。なお、前記クライアント情報処理装置１２０とのデータ通信をサーバプログラム１０９により実施し、前記中継プログラム１７０１は、必要に応じてサーバプログラム１０９が呼び出して利用するものであ

ってもよい。

【００８７】機密プログラム１７０３は、機密用ＯＳ１２２の上で動作するアプリケーションプログラムの１つであり、本実施例では、機密用ＯＳ１２２が占有するディスクコントローラ１１２ｃを利用して磁気ディスク１１４ｃに格納された機密情報１７０４にアクセスしたり、同じく機密用ＯＳ１２２が占有するＬＡＮコントローラ１１３ｃを経由して、占有する内部ネットワークに接続された記憶媒体中の機密情報１７０５にアクセスしながらサービスを提供するものと仮定して説明する。

【００８８】前記サーバ情報処理装置１００の起動時には、前記複数ＯＳ制御プログラム１１６が各ハードウェアの初期化、メモリ領域の分割、ＣＰＵのスケジューリング等を行う。

【００８９】次に、サービス用ＯＳ１０３とサーバプログラム１０９と中継プログラム１７０１がディスクコントローラ１１２ａを介して磁気ディスク１１４ａからメモリ１０１上にロードされ、セキュリティ用ＯＳ１０４とアクセス制御プログラム１１０がディスクコントローラ１１２ｂを介して磁気ディスク１１４ｂからメモリ１０２上にロードされ、更に機密用ＯＳ１２２と機密プログラム１７０３がディスクコントローラ１１２ｃを介して磁気ディスク１１４ｃからメモリ１７０８上にロードされる。

【００９０】図１８は、前記通信制御ポリシー１８００のデータ構造の一例を示したものである。アプリケーション名１８０１は、機密用ＯＳ１２２の上で動作する機密プログラムの名称であり、機密用ＯＳ１２２上でユニークな識別子となるよう、例えばプログラムファイルの絶対パス名等を用いる。サービス番号１８０２は、前記アプリケーション名１８０１に対応する機密プログラムが提供するサービスの識別子であり、例えばＴＣＰ／ＩＰ通信におけるポート番号や、メッセージ通信におけるメッセージキュー番号（メッセージキュー名称）等に相当する。本実施の形態では、１つの機密プログラムに対して１つのサービス番号を登録する場合を例に説明するが、これに限ったものではなく、複数のサービス番号を登録してもよい。サービス番号は、通信する双方で事前に取り決めておくものとし、中継プログラム１７０１から通信制御ポリシー１８００の参照はできないものとする。

【００９１】許可アプリケーション名１８０３は、サービス用ＯＳ１０３上で動作するプログラムであり、且つ前記アプリケーション名１８０１で示される機密プログラムとの通信を許可されたプログラムの名称であり、例えば中継プログラム１７０１に相当する。特徴値１８０４は、前記許可アプリケーション名１８０３で示されるプログラムが備える特徴値であり、例えば当該プログラムファイルのサイズやハッシュ値等に相当する。ユーザー名１８０５は、前記許可アプリケーション名１８０３で示されるプログラムが実行中に使用する権限をユーザー名で表したものである。これは、一般的なマルチユーザー・マルチタスクのＯＳにおいて、アプリケーションプログラムが必ず何れかのユーザー権限を使用して動作することを前提としている。特にユーザー名を限定したくない場合は、ユーザー名を所定の文字、記号（たとえばアスタリスク（＊））で表記する。なお、本通信制御ポリシー１８００は、システムの運用に先立って、管理者が予め登録しておくものである。また、図１８では機密プログラムと中継プログラムの組み合わせが１対１の場合を例に記述しているが、１つの機密プログラムの利用許可を、複数の中継プログラムに与えても良い。

【００９２】図１９は、前記通信ログ１９００のデータ構造の一例を示したものである。通信ログ１９００は、前記通信制御ポリシー１８００に違反する通信が発生した事実を書き込むためのファイルであり、当該通信が発生した日時１９０１と、当該通信の対象となった機密プログラム名１９０２と、機密プログラムとの通信要求を発行したサービス用ＯＳ１０３側プログラムを表すサブジェクト名１９０３、そしてエラーの内容を表すエラー番号１９０４から構成される。

【００９３】図２０は、通信管理テーブル２０００のデータ構造の一例を示したものである。通信管理テーブル２０００は、機密プログラムの識別子２００１と、当該機密プログラムが提供するサービスの番号２００２と、機密プログラムとのプロセス間通信のために使用する暗証データ２００３と、サービス用ＯＳ１０３のドライバ部１１７が提供する通信中のプログラムの識別子２００４、とから構成される。図２０では、各識別子にそれぞれプログラム名称を用いているが、例えば機密用ＯＳ１２２とサービス用ＯＳ１０３が各々管理するプロセスＩＤのような、プロセス特有の情報と組み合わせて管理してもよい。

【００９４】図２１から図２５は、中継プログラム１７０１と機密プログラム１７０３が、前記ＯＳ間通信処理部１０８の通信機能を利用してプロセス間通信をする際の、処理フローを示したものである。本実施の形態において、中継プログラム１７０１と機密プログラム１７０３は、クライアントとサーバの関係にあることから、前もって機密プログラム１７０３がデータ受信可能な状態になるものとして説明する。また、プロセス間通信手段には様々な種類があるが、どの手段を用いた場合でもその手順は、ハンドルの取得処理、データ送信処理とデータ受信処理、ハンドルの解放処理に大別できる。ハンドルとは、アプリケーションプログラムがファイル、メモリなどのオブジェクトをアクセスする際にアクセス対象となるオブジェクトを指定するデータであり，ＯＳから与えられる。ハンドルは、オブジェクトごとにシステム内でユニークな数値であることが保証されている。この場合のオブジェクトとは、例えばファイルやソケットな

ど、データ処理の際に用いるアクセス対象全般を指す。本発明では、機密用ＯＳ１２２やサービス用ＯＳ１０３上のプログラムがハンドル値を取得する際に、各ＯＳのドライバ部を経由して必ず通信制御プログラム１７００によるポリシーチェックを受け，正当性を確認するところに特徴がある。

【００９５】図２１は、前記ＯＳ間通信処理部１０８を介して、機密プログラム１７０３が通信制御プログラム１７００に対してサービスを登録し、中継プログラム１７０１からのデータ受信待ち状態になるまでの処理フローを示したものである。ステップ２１００にて、前記通信制御プログラム１７００は、ＯＳ間通信処理部１０８の機能を用いて機密用ＯＳ１２２側からのデータ受信が可能な状態となっている。機密プログラム１７０３は、ステップ２１０１にて暗証データを作成し、ステップ２１０２にて、機密用ＯＳ１２２のドライバ部１１９に対してサービス番号と暗証データを送付し、中継プログラム１７０１とのプロセス間通信に利用するハンドルを要求する。

【００９６】機密用ＯＳ１２２のドライバ部１１９は、ステップ２１０３にて通信制御プログラム１７００との通信路を確立し、ステップ２１０４にて前記機密プログラム１７０３の識別子とサービス番号と暗証データを、通信制御プログラム１７００に送付する。

【００９７】通信制御プログラム１７００は、ステップ２１０５にて通信制御ポリシー１８００を参照し、機密プログラムとサービス番号が登録されているかどうかを調べ，正当性を確認する。通信制御ポリシー１８００に記述されていれば、ステップ２１０６にて前記機密プログラム１７０３の識別子情報とサービス番号と暗証データを、通信管理テーブル２０００に登録する。通信制御ポリシー１８００に記述されていなければ、ステップ２１０７にてその旨を前記ドライバ部１１９に返信するためのエラー設定を行う。ステップ２１０８では、通信制御ポリシーとの照合結果を返信するとともに、ドライバ部１１８を介して、ドライバ部１１９との通信路を解放する。

【００９８】前記ドライバ部１１９は、ステップ２１０９にて通信制御プログラム１７００からの返信データをチェックし、通信管理テーブル２０００への登録に成功していれば、ステップ２１１１にて中継プログラム１７０１と通信するために利用するハンドルをＯＳ間通信処理部１０８から取得し、ステップ２１１２にて前記機密プログラム１７０３への返信データとして、前記ハンドル値を設定する。一方、通信管理テーブルへの登録に失敗した場合は、ステップ２１１０にて、前記機密プログラム１７０３への返信データとしてエラーを設定する。

【００９９】機密プログラム１７０３は、ステップ２１１３にてドライバ部１１９からの返信データをチェックする。要求したハンドル取得に成功していれば、ステップ２１１４にてサービス用ＯＳ１０３側からのデータ受信待ち状態になり、ハンドル取得に失敗していれば、ステップ２１１５にて所定のエラー処理を実行する。

【０１００】図２２は、前記機密プログラム１７０３がプロセス間通信に利用していたハンドルを解放する際の処理フローを示したものである。ステップ２２００にて、前記通信制御プログラム１７００は、ＯＳ間通信処理部１０８の機能を用いて機密用ＯＳ１２２側からのデータ受信が可能な状態となっている。機密プログラム１７０３は、ステップ２２０１にて、機密用ＯＳ１２２のドライバ部１１９に対してハンドル値を指定し、当該ハンドルの解放を要求する。

【０１０１】機密用ＯＳ１２２のドライバ部１１９は、ステップ２２０２にて通信制御プログラム１７００との通信路を確立し、ステップ２２０３にて前記機密プログラム１７０３の識別子とサービス番号を、通信制御プログラム１７００に送付する。

【０１０２】通信制御プログラム１７００では、ステップ２２０４にて通信管理テーブル２０００を参照し、登録された機密プログラムであるかを確認する。通信管理テーブル２０００に記述されていれば、ステップ２２０５にて前記機密プログラムの情報を、通信管理テーブル２０００から削除する。通信管理テーブル２０００に記述されていなければ、ステップ２２０６にてその旨を前記ドライバ部１１９に返信するためのエラー設定を行う。ステップ２２０７では、通信管理テーブル２０００との照合結果を返信するとともに、ドライバ部１１９との通信路を解放する。

【０１０３】前記ドライバ部１１９は、ステップ２２０８にて通信制御プログラム１７００からの返信データをチェックし、通信管理テーブルからの削除に成功していれば、ステップ２２１０にて今度は中継プログラム１７０１との通信に利用していたハンドルの解放処理を実行する。一方、通信管理テーブルからの削除に失敗した場合は、ステップ２２０９にて、前記機密プログラム１７０３への返信データとしてエラーを設定する。

【０１０４】機密プログラム１７０３は、ステップ２２１１にてドライバ部１１９からの返信データをチェックし、ハンドル解放に成功していれば、ステップ２２１２にて処理を終了する。一方、ハンドル解放に失敗していれば、ステップ２２１３にて所定のエラー処理を実行する。

【０１０５】図２３と図２４は、前記ＯＳ間通信処理部１０８を介して、中継プログラム１７０１が機密プログラム１７０３とのプロセス間通信用のハンドルを取得するまでの処理フローを示したものである。図２３のステップ２３００にて、前記通信制御プログラム１７００は、ＯＳ間通信処理部１０８の機能を用いてサービス用ＯＳ１０３側からのデータ受信が可能な状態となっている。中継プログラム１７０１は、ステップ２３０１に

て、サービス用ＯＳ１０３のドライバ部１１７に対して利用したいサービス番号を送付し、機密プログラムとのプロセス間通信に利用するハンドルを要求する。サービス番号は、上述の通り、ポート番号と同様に、事前に取り決めておくものとする。参照のために、プログラムに固定的に埋め込んでおくか、定義ファイルに記述しておくなどが考えられる。

【０１０６】サービス用ＯＳ１０３のドライバ部１１７は、ステップ２３０２にて通信制御プログラム１７００との通信路を確立し、ステップ２３０３にて前記中継プログラム１７０１の識別子とその特徴値、および中継プログラム１７０１が使用しているユーザー権限を表すユーザー名を取得し、利用したいサービス番号と共に通信制御プログラム１７００に送付する。

【０１０７】通信制御プログラム１７００では、ステップ２３０４にて通信管理テーブル２０００と通信制御ポリシー１８００を参照し、指定されたサービスが有効であり、且つ当該サービスの利用を許可されたプログラムか否かを調べ，正当性を確認する。具体的には、ドライバ部１１７から渡された中継プログラムの識別子と特徴値とユーザー名とが、通信制御ポリシー１８００の内容に一致すればポリシーに合致していると言える。もしサービスが有効で、且つポリシーに合致していれば、ステップ２３０５にて前記中継プログラム１７０１の識別子情報を、前記サービス利用中のプログラムとして通信管理テーブル２０００に登録し、該当するサービスの利用に必要な暗証データを中継プログラムに返信するために、通信管理テーブル２０００から取得する。一方、サービスが有効でない、あるいはポリシーに合致していなければ、ステップ２３０６にてその旨を通信ログ１９００に書き込むと共に、前記ドライバ部１１７に返信するためのエラー設定を行う。ステップ２３０７では、通信制御ポリシーとの照合結果を返信するとともに、ドライバ部１１７との通信路を解放する。

【０１０８】前記ドライバ部１１７は、ステップ２３０８にて通信制御プログラム１７００からの返信データをチェックし、通信管理テーブル２０００への登録に成功していれば、つまり中継プログラム１７０１の認証に成功していれば、ステップ２３１０にて今度は機密プログラムとの通信用ハンドルを取得する処理（図２４）に移る。一方、通信管理テーブルへの登録に失敗した場合は、ステップ２３０９にて、前記中継プログラム１７０１にエラーを返して終了する。

【０１０９】図２４では、図２１のステップ２１１４で示したように、前記機密プログラム１７０３がサービス用ＯＳ１０３側からデータ受信可能であることを前提にして、中継プログラム１７０１の通信用ハンドル取得処理を説明する。サービス用ＯＳ１０３のドライバ部１１７は、ＯＳ間通信処理部１０８の機能を利用して、機密プログラム１７０３との通信用ハンドルを取得する。ステップ２４０２では、先に図２３のステップ２３０７にて通信制御プログラム１７００から受信した暗証データを、機密用ＯＳ１２２のドライバ部１１９を経由して機密プログラム１７０３に送付する。

【０１１０】機密プログラム１７０３では、ステップ２４０３にて、ドライバ部１１７から受信した暗証データと、機密プログラム１７０３が作成した（図２１のステップ２１０１）暗証データとを照合し、結果を前記ドライバ部１１７に返信する。

【０１１１】前記ドライバ部１１７は、ステップ２４０４にて機密プログラム１７０３からの返信データをチェックする。前記暗証データが一致している場合には、前記機密プログラム１７０３との通信を許可されたと判断し、ステップ２４０５にて、中継プログラム１７０１への返信データとして、機密プログラム１７０３との通信ハンドル値を設定する。一方、前記暗証データに誤りがあった場合は、ステップ２４０６にて、機密プログラム１７０３との通信用ハンドルを解放すると共に、前記中継プログラム１７０１への返信データとしてエラーを設定する。

【０１１２】中継プログラム１７０１は、ステップ２４０７にてドライバ部１１７からの返信データをチェックし、ハンドル取得に成功していれば、以後、ＯＳ間通信処理部１０８の機能を利用して機密プログラム１７０３とのデータ通信が可能となる（ステップ２４０８）。一方、ハンドル取得に失敗していれば、ステップ２４０９にて所定のエラー処理を実行する。

【０１１３】図２５は、前記中継プログラム１７０１が機密プログラムとのプロセス間通信に利用していたハンドルを解放する際の処理フローを示したものである。ステップ２５００にて、前記通信制御プログラム１７００は、ＯＳ間通信処理部１０８の機能を用いてサービス用ＯＳ１０３側からのデータ受信が可能な状態となっている。中継プログラム１７０１は、ステップ２５０１にて、サービス用ＯＳ１０３のドライバ部１１７に対してハンドル値を指定して、当該ハンドルの解放を要求する。

【０１１４】サービス用ＯＳ１０３のドライバ部１１７は、ステップ２５０２にて通信制御プログラム１７００との通信路を確立し、ステップ２５０３にて前記中継プログラム１７０１の識別子とサービス番号を、通信制御プログラム１７００に送付する。

【０１１５】通信制御プログラム１７００では、ステップ２５０４にて通信管理テーブル２０００を参照し、当該テーブルに登録された中継プログラムであるかを確認する。通信管理テーブル２０００に登録されていれば、ステップ２５０５にて前記中継プログラムに該当する情報を、通信管理テーブル２０００から削除する。通信管理テーブル２０００に記述されていなければ、ステップ２５０６にてその旨を前記ドライバ部１１７に返信する

ためのエラー設定を行う。ステップ２５０７では、通信管理テーブル２０００との照合結果を返信するとともに、ドライバ部１１７との通信路を解放する。

【０１１６】前記ドライバ部１１７は、ステップ２５０８にて通信制御プログラム１７００からの返信データをチェックし、通信管理テーブルからの削除に成功していれば、ステップ２５１０にて今度は機密プログラム１７０３との通信に利用していたハンドルの解放処理を実行する。一方、通信管理テーブルからの削除に失敗した場合は、ステップ２５０９にて、前記中継プログラム１７０１への返信データとしてエラーを設定する。

【０１１７】中継プログラム１７０１は、ステップ２５１１にてドライバ部１１７からの返信データをチェックし、ハンドル解放に成功していれば、ステップ２５１２にて処理を終了する。一方、ハンドル解放に失敗していれば、ステップ２５１３にて所定のエラー処理を実行する。

【０１１８】以上説明したように、本発明によれば、侵入者が強い権限をもつユーザーに成りすました場合でも、ファイルへのアクセスを制限できるという効果がある。

【０１１９】また、不正なファイルアクセスを未然に防ぐことができるという効果がある。

【０１２０】また、機密性の高い情報へのアクセス手段となるプログラム自体を保護できるという効果がある。また、前記ポリシーファイルとアクセス制御手段を、外部からの不正アクセスや攻撃から保護できるという効果がある。

【０１２１】また、図１に示すシステムの、サーバ情報処理装置とクライアント情報処理装置とを結ぶLAN１１５や、図１７に示すシステムの、サーバ情報処理装置とクライアント情報処理装置とを結ぶ外部ネットワーク１７０６などの通信回線上にファイアウォールを設け、さらに本発明を併用すれば、より強固なセキュリティを確保することが出来る。

【０１２２】

【発明の効果】本発明によれば、情報処理装置が管理するファイル、情報、および実行中のプロセスを不正なアクセスから保護することが可能になる。

【図面の簡単な説明】

【図１】本発明の実施の形態におけるアクセス制御システムの一構成例を示す図。

【図２】アクセス制御ポリシーの設定を格納するためのポリシーファイルを示す図。

【図３】不正アクセスが発生した事実を記録するためのアクセスログファイルを示す図。

【図４】ファイルＩ／Ｏフックプログラム１０６とアクセス制御プログラム１１０を構成するプログラムルーチンを示す図。

【図５】本発明の実施の一形態において、各モジュール間を流れるデータの通信経路を示す図。

【図６】本発明の実施の一形態において、各モジュール間を流れるデータの構造を示す図。

【図７】ファイルＩ／Ｏフックルーチン４００の処理のフローチャートを示す図。

【図８】オープン処理ルーチン４０１の処理のフローチャートを示す図。

【図９】オープン制御ルーチン４０５の処理のフローチャートを示す図。

【図１０】オープンファイルテーブルの構造体とそのリストを示す図。

【図１１】オープンファイルテーブルに登録されるデータの一例を示す図。

【図１２】クローズ処理ルーチン４０２の処理のフローチャートを示す図。

【図１３】リード・ライト処理ルーチン４０３の処理のフローチャートを示す図。

【図１４】アクセスログ登録ルーチン４０６の処理のフローチャートを示す図。

【図１５】削除・リネーム処理ルーチン４０４の処理のフローチャートを示す図。

【図１６】削除・リネーム処理ルーチン４０７の処理のフローチャートを示す図。

【図１７】プロセス間通信に関するアクセス制御システムの一構成例を示す図。

【図１８】プロセス間通信に関するポリシーの設定を格納するためのファイルを示す図。

【図１９】不正なプロセス間通信が発生した事実を記録するためのログファイルを示す図。

【図２０】通信管理テーブルのリストを示す図。

【図２１】機密用ＯＳ１２２上のプログラムによる通信用ハンドル取得処理のフローチャートを示す図。

【図２２】機密用ＯＳ１２２上のプログラムによる通信用ハンドル解放処理のフローチャートを示す図。

【図２３】サービス用ＯＳ１０３上のプログラムによる通信用ハンドル取得処理のフローチャート（前半）を示す図。

【図２４】サービス用ＯＳ１０３上のプログラムによる通信用ハンドル取得処理のフローチャート（後半）を示す図。

【図２５】サービス用ＯＳ１０３上のプログラムによる通信用ハンドル解放処理のフローチャートを示す図。

【図２６】オープン制御ルーチン４０５の処理のフローチャート（その２）を示す図。

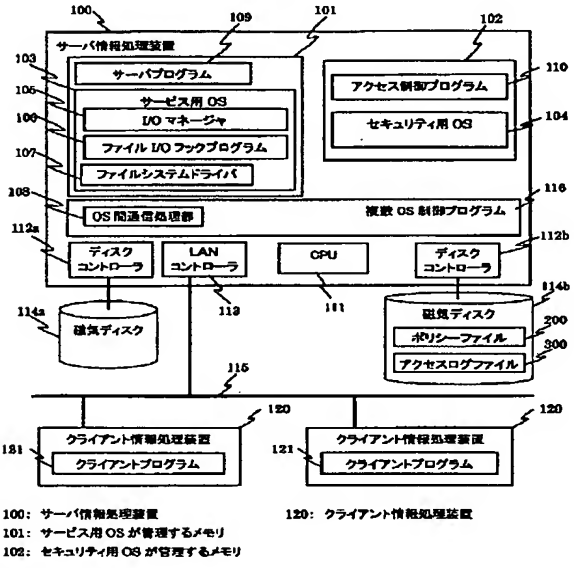
【符号の説明】

１００…サーバ情報処理装置、１０１…サービス用ＯＳが管理するメモリ、１０２…セキュリティ用ＯＳが管理するメモリ、１０３…サービス用ＯＳ、１０４…セキュリティ用ＯＳ、１０５…Ｉ／Ｏマネージャ、１０６…ファイルＩ／Ｏフックプログラム、１０７…フ

ファイルシステムドライバ、１０８･･･ＯＳ間通信制御部、１０９･･･サーバプログラム、１１０･･･アクセス制御プログラム、１１１･･･ＣＰＵ、１１２･･･ディスクコントローラ、１１３･･･ＬＡＮコントローラ、１１４･･･磁気デバイス、１１５･･･ＬＡＮ、１１６･･･複数ＯＳ制御プログラム、１１７～１１９･･･ドライバ部、１２０･･･クライアント情報処理装置、１２１･･･クライアントプログラム、２００･･･ポリシーファイル、３００･･･アクセスログファイル、５０１～５０６･･･通信経路、６００･･･Ｉ／Ｏパケット、６１０･･･リクエストパケット、６２０･･･レスポンスパケット、１０００･･･オープンファイルテーブル、１７００･･･通信制御プログラム、１７０１･･･中継プログラム、１７０３･･･機密プログラム、１７０４～１７０５･･･機密情報、１７０６･･･外部ネットワークとの接続回線、１７０７･･･内部ネットワークとの接続回線、１７０８･･･機密用ＯＳが管理するメモリ、１８００･･･通信制御ポリシー、１９００･･･通信ログ、２０００･･･通信管理テーブル

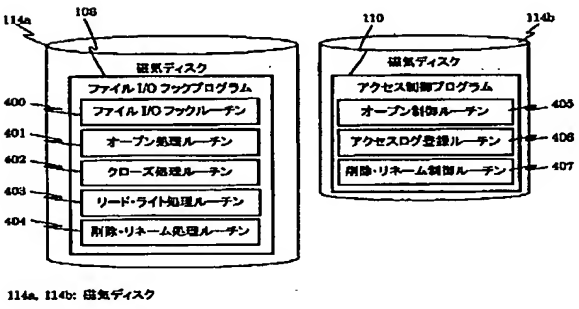
【図１】

図１



【図４】

図４



【図３】

図３

300

301 日時	302 オブジェクト名	303 アクセスタイプ	304 サブジェクト名	305 ユーザー名
1999.07.28 15:32:46	D:¥DOC¥SECRET.TXT	Write	c:¥prog¥wwwsrv.exe	u_0023
1999.07.27 09:15:10	D:¥DOC¥SECRET.TXT	Delete	c:¥prog¥wwwsrv.exe	u_0023
1999.07.25 16:02:55	D:¥SYS¥CONFIG¥	Write	c:¥prog¥txtedit.exe	intruder
1999.07.25 14:44:28	D:¥SYS¥CONFIG¥	Delete	c:¥prog¥fileman.exe	intruder
1999.07.25 14:42:59	D:¥LOG¥LOG.TXT	Read	c:¥prog¥txtedit.exe	user007
1999.07.16 10:29:31	D:¥LOG¥LOG.TXT	Delete	c:¥prog¥fileman.exe	user007

300: アクセスログファイル

【図２】

図２

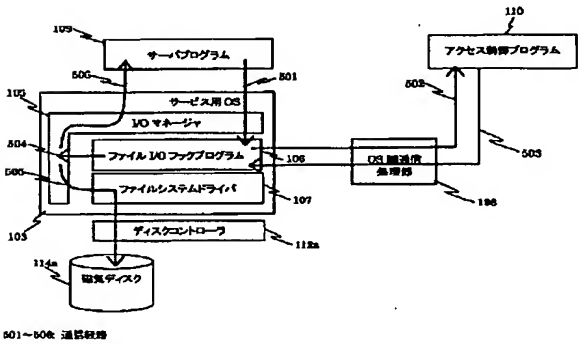
200

210 オブジェクト名	201 禁止されたアクセスのタイプ	202 エラーコード	203 例外サブジェクト	204 プログラムのハッシュ値	205 ユーザー名	206
D:¥DOC¥SECRET.TXT	Open	0016	c:¥prog¥wordedit.exe	0x22F0A412B73209CC	sec_admin	
	Delete	0021	c:¥prog¥fileman.exe	0x73209C2F0A212B4C	sec_admin	
211 D:¥SYS¥CONFIG¥*	Write	0018	c:¥prog¥mantool.exe	0xB74122209FDE236C	sys_admin	
	Delete	0021	c:¥prog¥mantool.exe	0xB74122209FDE236C	sys_admin	
212 D:¥LOG¥LOG.TXT	Read	0015	c:¥prog¥audit.exe, c:¥prog¥vck.exe	0x2F21204B73C09 A2C 0xF0A271243B9C202C	root, system	
	Rename	0024	c:¥prog¥audit.exe	0x2F21204B73C09 A2C	root	

200: ポリシーファイル

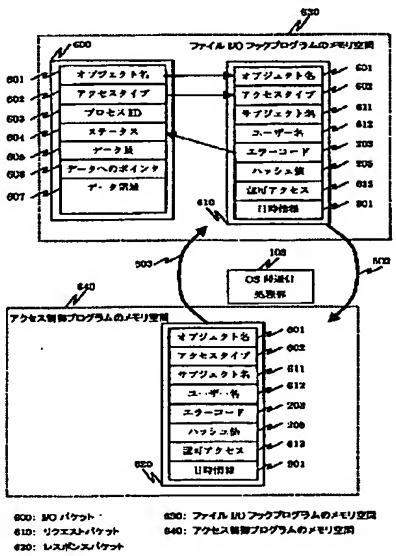
【図５】

図５



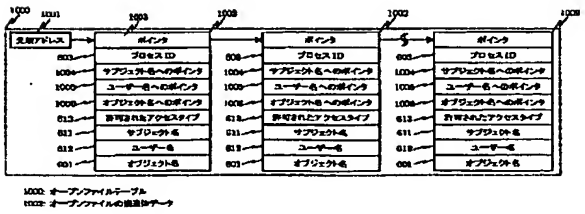
【図６】

図６



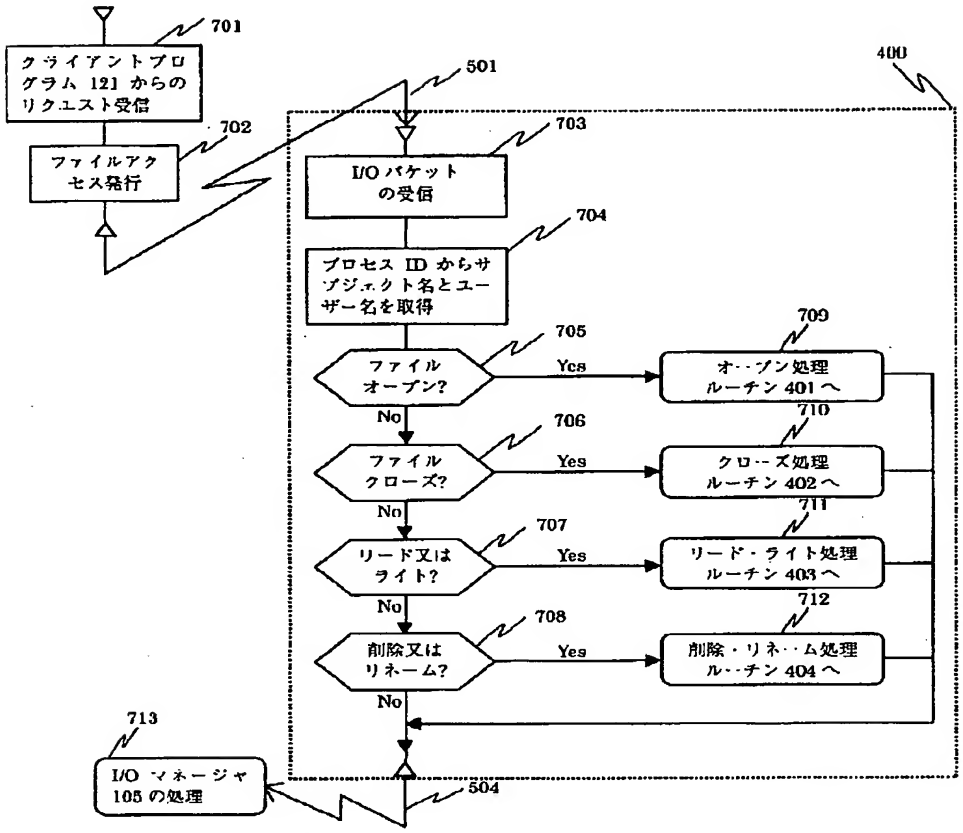
【図１０】

図１０



【図７】

図７



701～702: サーバプログラム 109 の概略処理フロー

703～712: ファイル I/O フックルーチン 400 の処理フロー

【図１１】

図１１

1000

603 プロセスID	611 サブジェクト名	612 ユーザー名	601 オブジェクト名	613 許可されたアクセスタイプ
0022	c:¥prog¥wwwserv.exe	inet	D:¥HOME¥SALES.HTML	Read
0043	c:¥prog¥wordedit.exe	sec_admin	D:¥DOC¥SECRET.TXT	Read/Write
0057	c:¥prog¥viewer.exe	tawou	D:¥DOC¥IDEA.TXT	Read
0092	c:¥prog¥mantool.exe	sys_admin	D:¥SYS¥CONFIG¥PORT	Read/Write
0113	c:¥prog¥vch.exe	system	D:¥DOC¥SHEET.DOC	Read
0218	c:¥prog¥audit.exe	root	D:¥LOG¥LOG.TXT	Read/Write

【図１９】

図１９

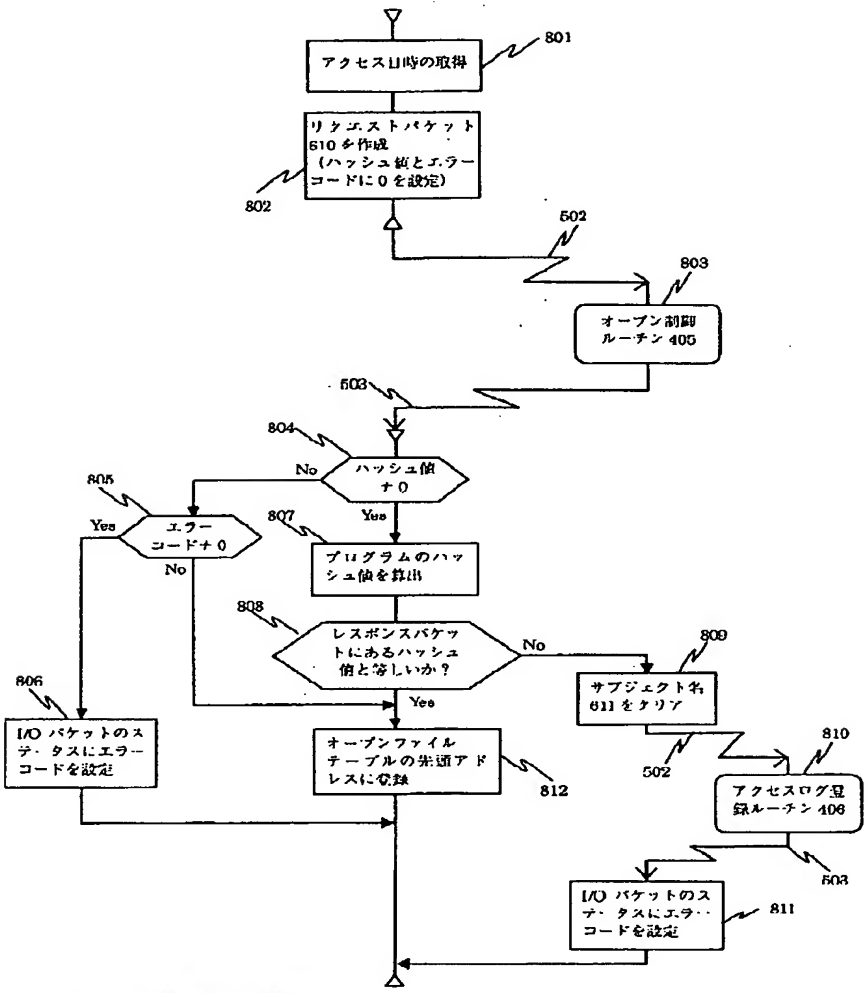
1900

1901 日時	1902 機密プログラム名	1903 サブジェクト名	1904 エラー番号
2000.06.23 16:13:49	/bin/prog/login	c:¥prog¥cgi¥login.exe	0x00CC
2000.06.29 03:44:06	/bin/prog/sasdaka	c:¥prog¥cgi¥zand.exe	0x0099
2000.07.05 03:13:12	/bin/prog/shinsei	c:¥prog¥cgi¥shin.exe	0x00A0

1900: 通信ログ

【図８】

図８



801～810: オープン処理ルーチンの処理フロー

【図１８】

図１８

1800

1801 アプリケーション名	1802 サービス番号	1803 許可アプリケーション名	1804 特徴量	1805 ユーザー名
/bin/prog/login	30	c:¥prog¥cgi¥login.exe	0xA432B7CC	www
/bin/prog/zandaka	80	c:¥prog¥cgi¥zand.exe	0x2875FA99	www
/bin/prog/shinsei	100	c:¥prog¥cgi¥shin.exe	0x5F1F66A6	www

1800: 通信制御ポリシー

【図２０】

図２０

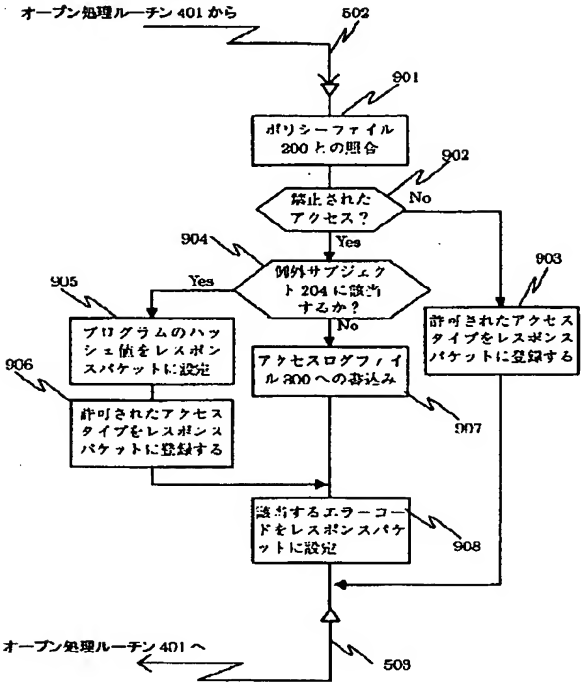
2000

2001 機密プログラムの識別子	2002 サービス番号	2003 暗証データ	2004 通信中プログラムの識別子
/bin/prog/login	30	0x228H	c:¥prog¥cgi¥login.exe
/bin/prog/zandaka	80	0x9A0C	
/bin/prog/shinsei	100	0x3175	

2000: 通信管理テーブル

【図９】

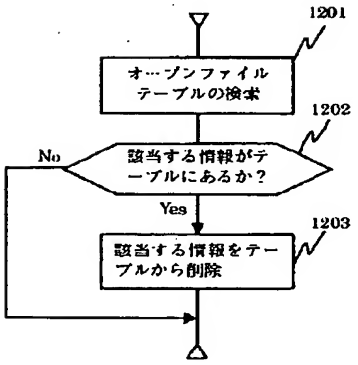
図９



901～908: オープン制御ルーチン 405 の処理フロー

【図１２】

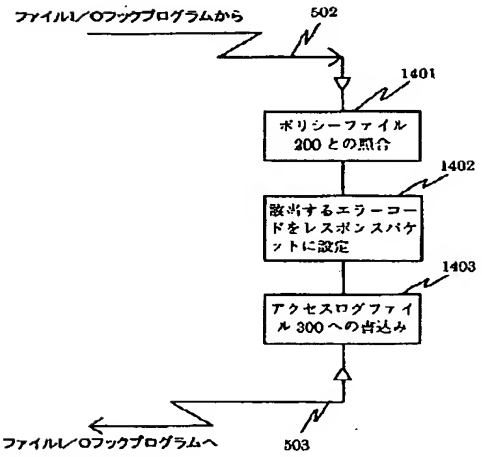
図１２



1201～1203: クローズ処理ルーチン 402 の処理フロー

【図１４】

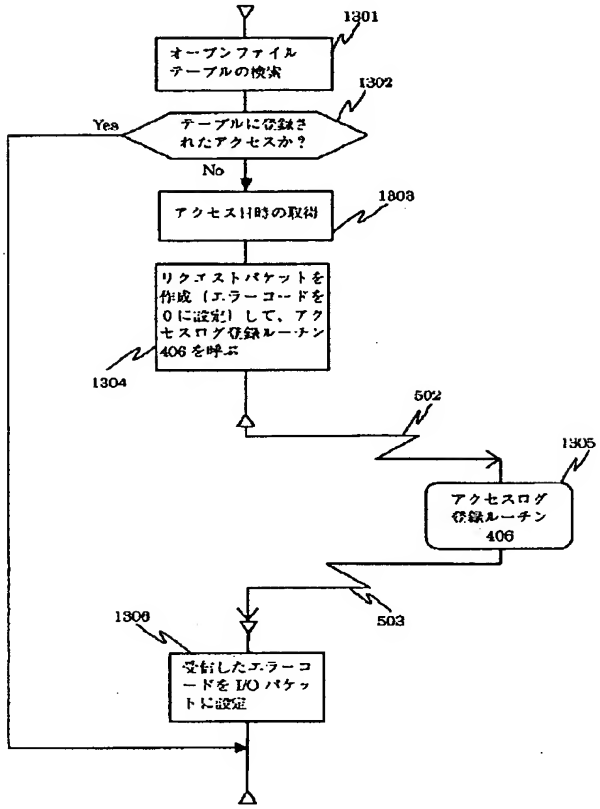
図１４



1401～1403: アクセスログ登録ルーチン 406 の処理フロー

【図１３】

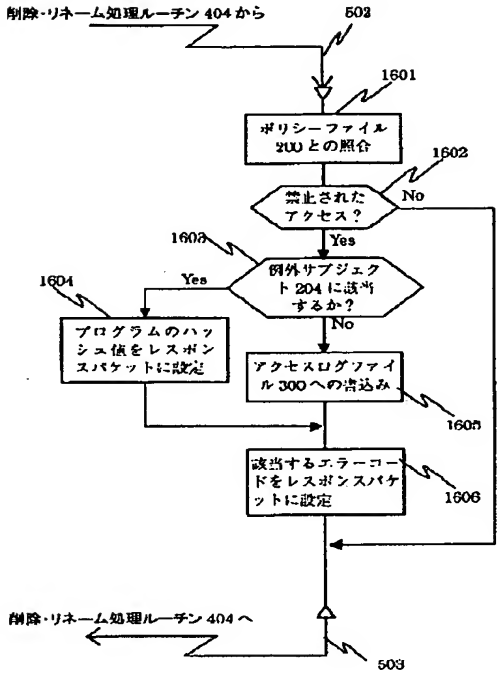
図１３



1301～1306: リード・ライト処理ルーチン 403 の処理フロー

【図１６】

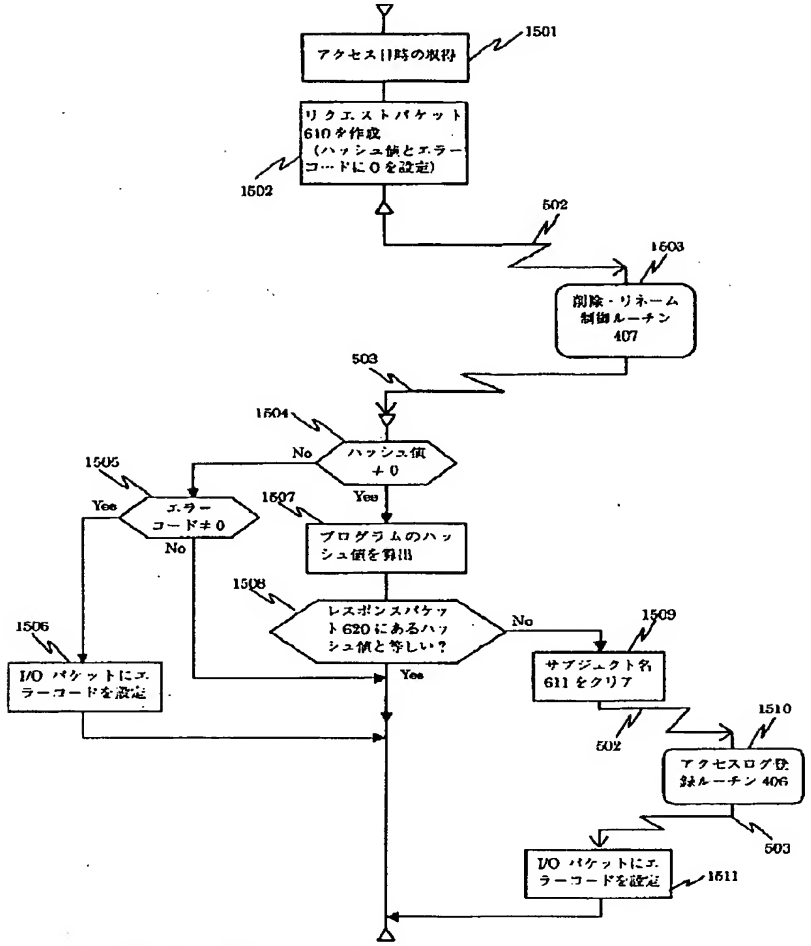
図１６



1601～1606: 削除・リネーム制御ルーチン 407 の処理シーケンス

【図１５】

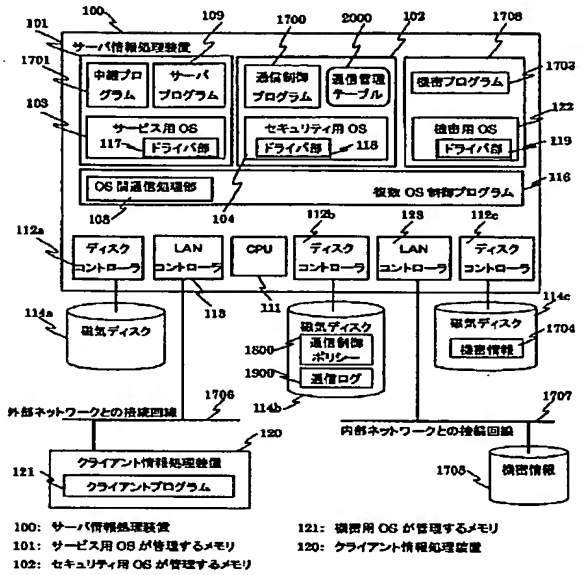
図１５



1501～1509: 削除・リネーム処理ルーチン 404 の処理フロー

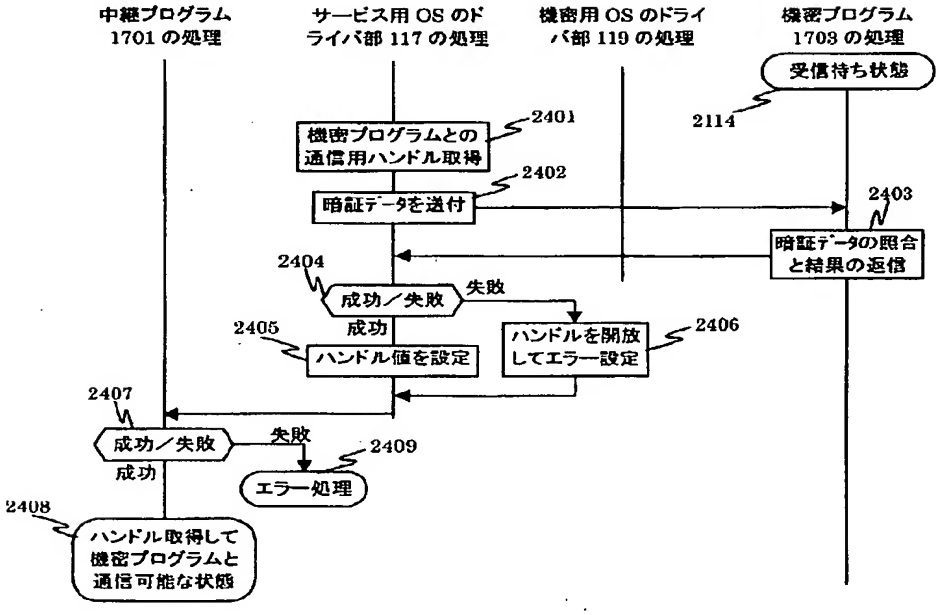
【図１７】

図１７



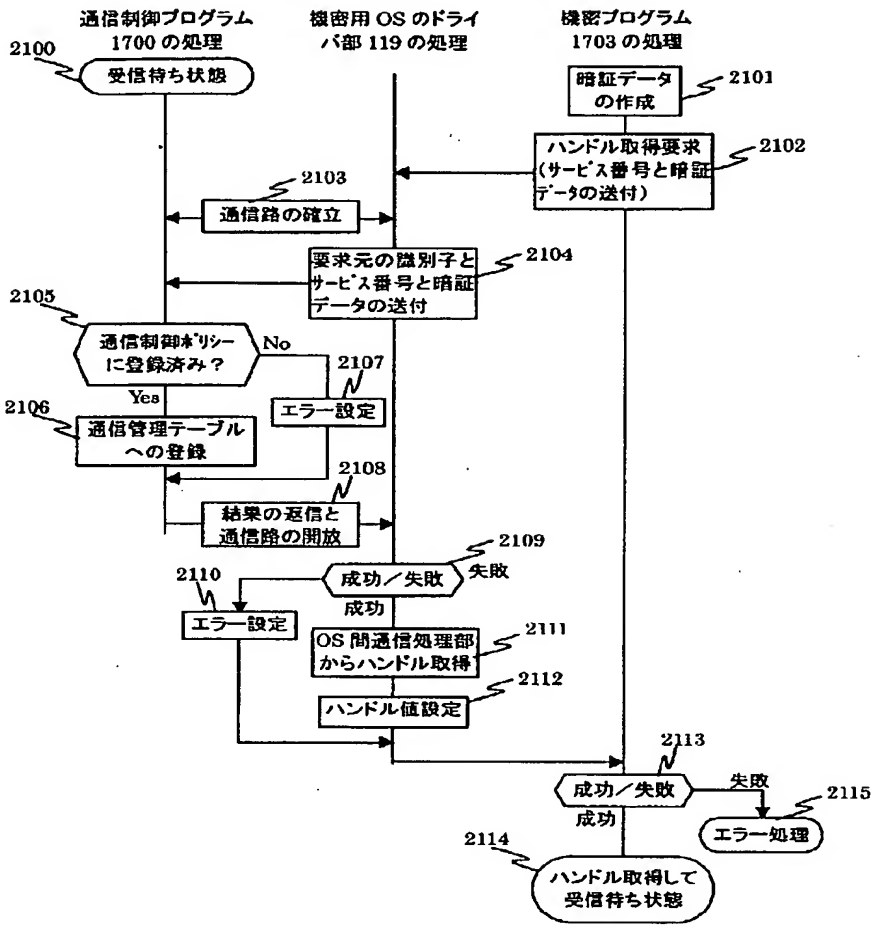
【図２４】

図２４



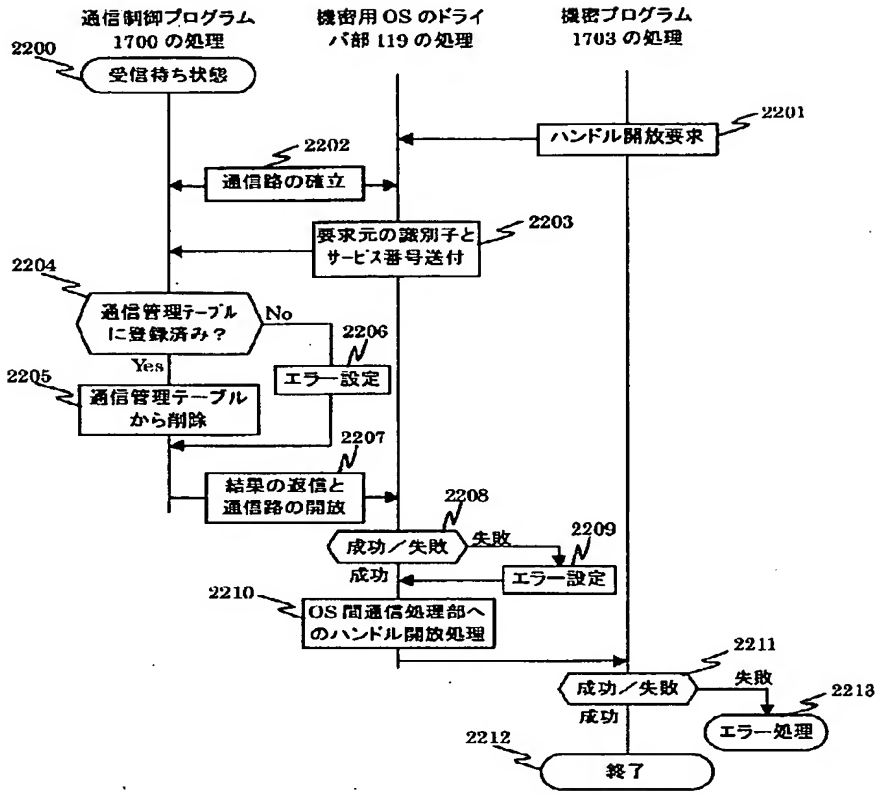
【図２１】

図２１


通信制御プログラム
1700 の処理
機密用 OS のドライバ部 119 の処理
機密プログラム
1703 の処理
2100
受信待ち状態
暗証データの作成
2101
ハンドル取得要求（サービス番号と暗証データの送付）
2102
2103
通信路の確立
要求元の識別子とサービス番号と暗証データの送付
2104
2105
通信制御ポリシーに登録済み？
No
2107
エラー設定
Yes
2106
通信管理テーブルへの登録
2108
結果の返信と通信路の開放
2109
成功／失敗
失敗
2110
エラー設定
成功
OS 間通信処理部からハンドル取得
2111
ハンドル値設定
2112
2113
成功／失敗
失敗
2115
エラー処理
成功
2114
ハンドル取得して受信待ち状態

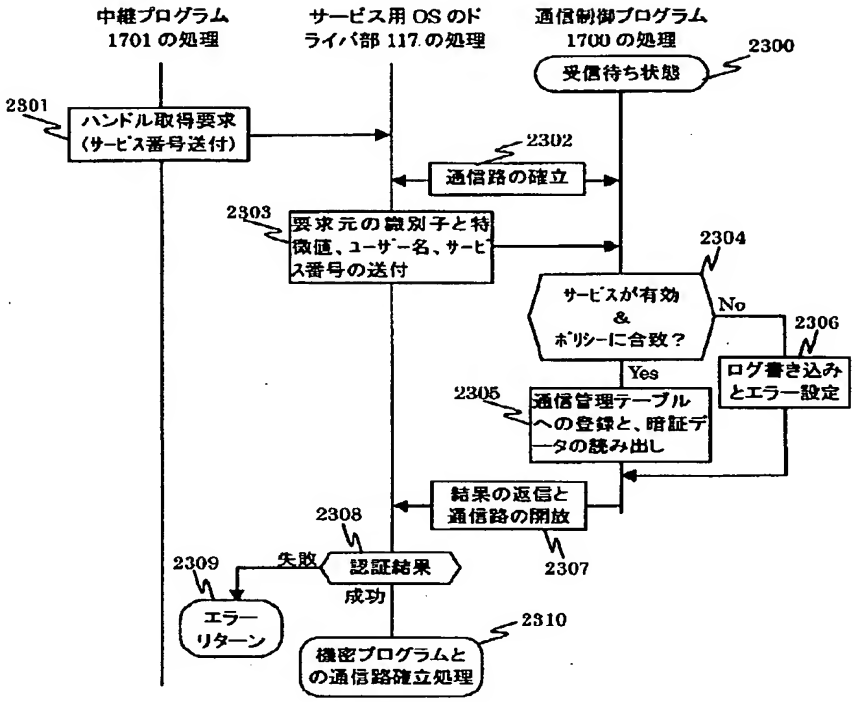
【図２２】

図２２


通信制御プログラム
1700 の処理
機密用 OS のドライバ部 119 の処理
機密プログラム
1703 の処理
2200
受信待ち状態
2201
ハンドル開放要求
2202
通信路の確立
2203
要求元の識別子とサービス番号送付
2204
通信管理テーブルに登録済み？
No
Yes
2205
通信管理テーブルから削除
2206
エラー設定
2207
結果の返信と通信路の開放
2208
成功／失敗
失敗
2209
エラー設定
成功
2210
OS 間通信処理部へのハンドル開放処理
2211
成功／失敗
失敗
2213
エラー処理
成功
2212
終了

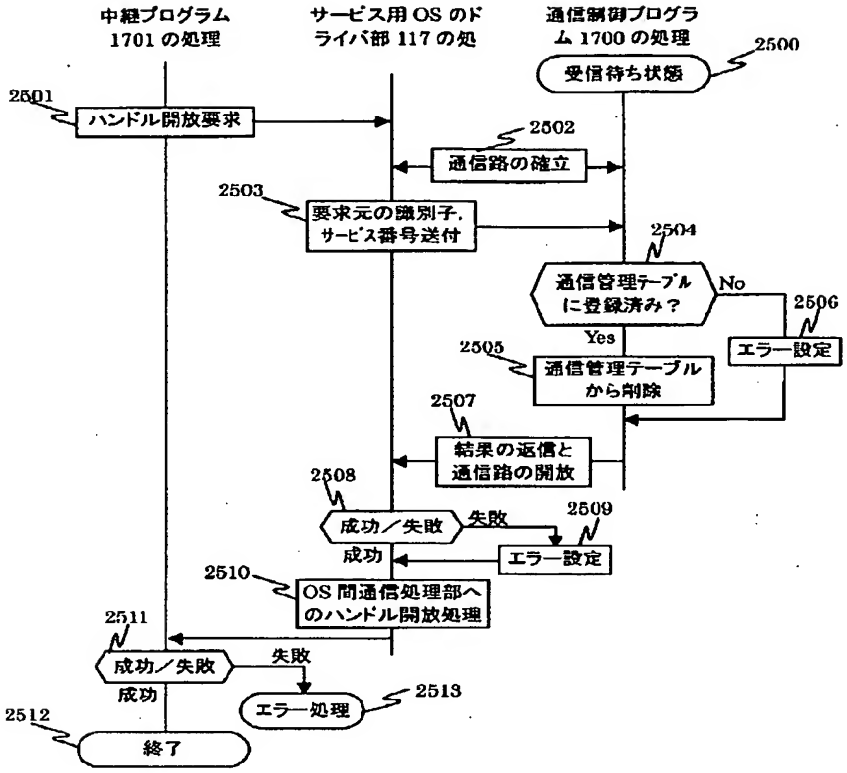
【図２３】

図２３


中継プログラム
1701 の処理
サービス用 OS のド
ライバ部 117 の処理
通信制御プログラム
1700 の処理
2300
受信待ち状態
2301
ハンドル取得要求
(サービス番号送付)
2302
通信路の確立
2303
要求元の識別子と特
徴値、ユーザー名、サービ
ス番号の送付
2304
サービスが有効
&
ポリシーに合致？
No
2306
ログ書き込み
とエラー設定
Yes
2305
通信管理テーブル
への登録と、暗証デ
ータの読み出し
結果の返信と
通信路の開放
2307
2308
失敗
認証結果
2309
成功
エラー
リターン
2310
機密プログラムと
の通信路確立処理

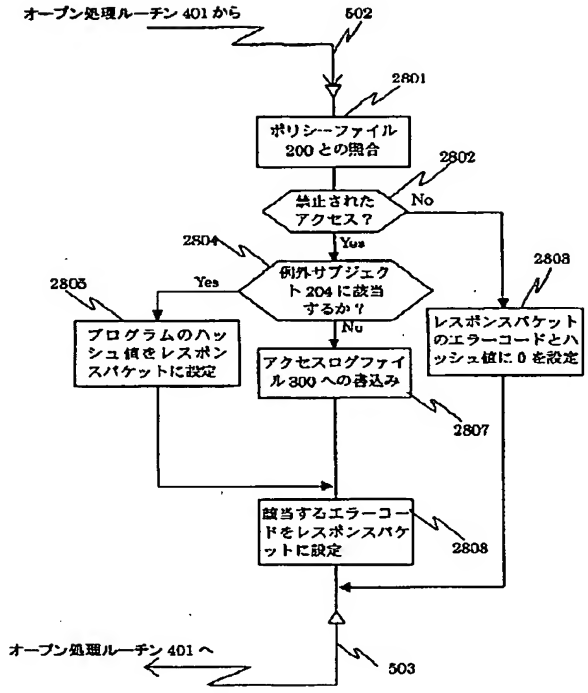
【図２５】

図２５


中継プログラム 1701 の処理
サービス用 OS のドライバ部 117 の処
通信制御プログラム 1700 の処理
2500
受信待ち状態
2501
ハンドル開放要求
2502
通信路の確立
2503
要求元の識別子，サービス番号送付
2504
通信管理テーブルに登録済み？
No
2506
エラー設定
Yes
2505
通信管理テーブルから削除
2507
結果の返信と通信路の開放
2508
成功／失敗
失敗
2509
エラー設定
成功
2510
OS 間通信処理部へのハンドル開放処理
2511
成功／失敗
失敗
成功
エラー処理
2513
2512
終了

【図２６】

図２６



2801～2808: オープン制御ルーチン 405 の処理フロー

フロントページの続き

(72)発明者　梶浦　敏範
愛知県尾張旭市晴丘町池上１番地　株式会社日立製作所情報機器事業部内

35　Ｆターム(参考)　5B017 AA01 BA06 BB06 CA16
5B076 FB05
5B082 GA11
5B085 AE00 BG03 BG07